

Niveau(x) d'enseignement: 4
Centre d'intérêt: 640

Numéro EDUQ : 8670296

St-Cyr, Pierre.

Le perfectionnement du personnel scolaire en micro-informatique : guide de formation pour un programme de 5 jours à l'intention des enseignantes et des enseignants du cours ISI / auteur, Pierre St-Cyr ; coord., Claude Labrecque ; [pour la Direction générale des régions, Services éducatifs] , -- [Québec] : La Direction, c1986.
207 p. : graph. ; 28 cm.

Code: 51-5067. Bibliogr. : p. 183-184.

'9BN 2-550-08478-0

Vedette(s)-secondaire(s): Labrecque, Claude, éd * Québec (Province)
Direction des services éducatifs

Descripteur(s): Micro-informatique * Initiation à l'informatique * Perfectionnement des enseignants * Guide pédagogique * Secondaire (Ordre d'enseignement)

Le guide de formation à l'intention des enseignants du cours "d'Introduction à la Science de l'informatique" (ISI) vise à faire connaître aux participants des stages la didactique du programme qui est centrée sur l'approche-projet. Deux objectifs sont poursuivis: 1. permettre aux participants de se situer face au modèle présenté et à la philosophie qui l'anime; 2. amener les participants à réajuster leur intervention et à modifier leur approche pédagogique. Le guide comprend cinq modules: 1. la résolution des problèmes; 2. l'élaboration des projets; 3. la gestion des projets; 4. la planification de l'année; 5. l'évaluation.

Type de document: d

LE PERFECTIONNEMENT DU PERSONNEL SCOLAIRE EN MICRO-INFORMATIQUE

**GUIDE DE FORMATION
POUR UN PROGRAMME DE 5 JOURS
À L'INTENTION DES ENSEIGNANTES ET DES ENSEIGNANTS
DU COURS ISI**

Québec ☐☐

LE PERFECTIONNEMENT DU PERSONNEL SCOLAIRE EN MICRO-INFORMATIQUE

**GUIDE DE FORMATION
POUR UN PROGRAMME DE 5 JOURS
À L'INTENTION DES ENSEIGNANTES ET DES ENSEIGNANTS
DU COURS ISI**

Québec ☐☐

**GUIDE DE FORMATION
POUR UN PROGRAMME DE 5 JOURS
À L'INTENTION DES ENSEIGNANTES ET DES ENSEIGNANTS
DU COURS ISI**



AUTEUR: PIERRE ST-CYR

Coordination: Claude Labrecque

**Direction générale des régions
Services éducatifs**

PROLOGUE

À la fin du cours secondaire général, les étudiants et les étudiantes ont la possibilité de choisir l'option "Introduction à la Science de l'Informatique" communément appelée "cours I.S.I.".

Depuis l'annonce de l'implantation de ce programme d'études jusqu'à son implantation réelle, le Ministère de l'Éducation s'est préoccupé de la préparation des enseignants et des enseignantes à une animation adéquate de ce cours. Diverses démarches ont été entreprises pour offrir du perfectionnement aux intéressés.

En cette année scolaire 1985-86, les Services Éducatifs de la Direction Générale des Régions ont offert aux Commissions scolaires un programme de stages de 2 à 5 jours pour le personnel rattaché au cours I.S.I. Concomitamment avec l'organisation de ces stages a été réalisée l'élaboration du présent "guide de formation".

Pierre St-Cyr, enseignant de la C.E.C.M. qui a animé ces stages, est l'auteur du présent guide.

La coordination des stages et de la publication du guide a été assurée par Claude Labrecque adjoint au coordonnateur en micro-informatique des Services Éducatifs de la D.G.R.

M.E.Q., D.G.R.
Services Éducatifs

TABLE DES MATIÈRES

Esprit du présent guide	1
Plan du guide	3
MODULE 1: La résolution de problèmes	5
MODULE 2: L'élaboration de projets	47
MODULE 3: La gestion de projets	87
MODULE 4: La planification de l'année	107
MODULE 5: L'évaluation	119
Bibliographie générale	183
Annexes	185

PRÉSENTATION DU GUIDE

ESPRIT DU PRÉSENT GUIDE

Ce guide n'a pas la prétention d'être la panacée à tous les problèmes que pose l'enseignement du cours d'Introduction à la Science de l'Informatique.

Les pages qui suivent suggèrent une approche axée sur l'acquisition d'une méthode de travail plutôt que sur l'apprentissage d'un langage de programmation. Le projet-étudiant devient une démarche importante, complète et personnelle vers l'autonomie. Réaliser un projet consiste à apprendre à chercher de l'information et à savoir organiser cette information. L'enseignant aide l'étudiant à structurer son travail. Il s'agit d'amener les élèves à "réfléchir sur papier" avant d'entreprendre le travail sur ordinateur.

Fortement inspirés des techniques de conception de logiciels (software design) nous suggérons des étapes d'élaboration de projet. Cette méthode de travail, assistée d'une bonne gestion de projet, semble indispensable pour résoudre un problème à l'aide de l'ordinateur.

BUTS ET OBJECTIFS DU GUIDE

Le guide de formation à l'intention des enseignantes et enseignants du cours ISI vise à faire connaître aux participants la didactique du programme qui est centrée sur l'approche-projet.

Deux objectifs sont poursuivis:

1. Permettre aux participants de se situer face au modèle présenté et à la philosophie qui l'anime.
2. Amener les participants à réajuster leur intervention et/ou modifier leur approche pédagogique.

INTRODUCTION

INTRODUCTION

Avant l'avènement des langages évolués, la machine n'acceptait que le binaire et rendait alors la tâche de codification difficile et laborieuse. L'année 1957 rend disponible le Fortran. Le chercheur s'est alors passionné pour le langage oubliant tout à fait que celui-ci le rendait plus libre qu'auparavant pour réfléchir. Les années qui suivent nous apportent une multitude de langages avec chacun des avantages et des désavantages. Comme le souligne Jacques Arsac penseur et pédagogue français, on a tout à fait oublié que l'essentiel n'est pas de disposer d'un langage mais d'avoir quelque chose à dire. Enseigner un langage de programmation ou faire de la syntaxe n'est pas enseigner une méthode de travail. La tentation est forte pour le professeur! Il y a de plus en plus de microordinateurs dans les laboratoires et les élèves ne demandent pas mieux que de passer des périodes entières sur les appareils.

Le cours d'Introduction à la Science de l'Informatique propose à l'étudiant une méthode de travail. Les étudiants(es) s'approprieront donc une méthode propre à la science de l'informatique. Il faudra développer des algorithmes, découvrir des stratégies de résolution de problèmes acceptables et utilisables par l'ordinateur. Après l'approche sauvage des premiers cours, les étudiants comprendront vite qu'il est nécessaire et primordial de prendre un certain recul face à l'appareil pour mieux réfléchir. Il faut s'interroger sur le travail à faire, analyser le problème à résoudre, formuler des hypothèses, inventorier des stratégies de résolution.

Comme le souligne Mike James dans son livre "Mieux Programmer", pour programmer il faut connaître un langage que l'ordinateur peut comprendre, mais cette connaissance ne veut pas nécessairement dire que l'on sait pour autant programmer. La connaissance d'un langage et avoir des idées sont deux choses différentes. L'expérience et la pratique de la résolution de problèmes permettront d'apprendre à bien programmer. Il y a des techniques de résolution qu'on apprend avec le temps, qu'on expérimente à plusieurs occasions et qui sont indépendantes des langages de programmation.

PLAN DU GUIDE

<u>LES MODULES</u>	<u>TEMPS</u>
1. La résolution de problèmes	1 j.
2. L'élaboration de projets	2 j.
3. La gestion de projets	1/2 j.
4. La planification de l'année (enseignement et évaluation)	1/2 j.
5. L'évaluation	1 j.

MODULE 1

LA RÉOLUTION DE PROBLÈMES

OBJECTIF TERMINAL

S'approprier des outils utiles à l'élaboration de projets.

OBJECTIFS INTERMÉDIAIRES

- 1.1 Connaître certaines techniques de résolution de problèmes
- 1.2 Utiliser certaines techniques de résolution de problèmes
- 1.3 Utiliser des outils de conception
- 1.4 Élaborer des exercices et/ou des problèmes en regard de l'objectif général 2 du programme d'études

CONTENU

- A. Techniques de résolution de problèmes
- B. Exercices de résolution de problèmes
- C. Exemples d'utilisation de l'ordinogramme, du pseudo-langage, du diagramme de Warnier-Orr et des arbres algorithmiques.

DURÉE SUGGÉRÉE: 1 jour

INSTRUMENTATION

A. ACTIVITÉ D'APPRENTISSAGE

- Techniques de résolution de problèmes
- Représentations graphiques
- Utilisation du pseudo-langage

B. MATRICES DE TRANSPARENTS

- Figures encadrées susceptibles d'être reproduites à des fins pédagogiques

C. MATÉRIEL DIDACTIQUE

- L'ordinogramme en Annexe I du présent document
- Le pseudo-langage en Annexe II du présent document
- L'arbre algorithmique en Annexe III du présent-document

D. DOCUMENTATION COMPLÉMENTAIRE

- Bibliographie générale

RÉSOLUTION DE PROBLÈMES

1.1 Problématique

La résolution d'un problème, à l'aide de l'ordinateur, tient compte de trois composantes initiales d'une démarche propre à la science de l'informatique:

- l'entrée des données: quelles sont les choses que l'ordinateur doit connaître pour passer à l'exécution?
- le traitement: qu'est-ce qui doit être fait?
- la sortie des résultats: quel sera le produit final?

La résolution des problèmes est la recherche de la façon d'aller d'une situation initiale à un but désiré. C'est une activité mentale du type "comprendre comment".

1.2 Méthode générale et interventions

Après avoir défini de façon précise les entrées et les sorties, il faudra développer l'algorithme de résolution.

Lors de la conception d'un programme informatique intervient un savoir-faire indispensable. Ce savoir-faire consiste en un ensemble de méthodes disparates, mal définies et floues qu'on appelle heuristiques. Les heuristiques sont des mécanismes spécifiques mis en place lors de la période d'exploration-découverte. Ces méthodes sont indispensables lors de la simulation d'un raisonnement et permettent ainsi de restreindre l'espace des recherches aux seules directions vraisemblables.

Voici une liste de quelques-unes de ces actions utilisées au cours de la résolution de problèmes:

- | | | |
|---------------|----------------|---------------|
| - éliminer | - chercher | - transformer |
| - associer | - sélectionner | - traduire |
| - classifier | - planifier | - développer |
| - généraliser | - prédire | - réduire |
| - comparer | - questionner | - exagérer |
| - dégager | - conjecturer | - amoindrir |
| - imaginer | - deviner | - adapter |

L'élève choisit les heuristiques qui lui conviennent selon son imagination, son expérience antérieure et ses compétences.

A l'aide de ses connaissances, de ses habiletés, de son expérience, de sa créativité, de la collaboration de ses confrères ou d'autres personnes, l'élève essaie de découvrir, d'une manière autonome, une solution à son problème.

Tel que mentionné précédemment la résolution de problèmes vise à développer chez l'étudiant l'autonomie intellectuelle i.e.

- trouver des stratégies de résolution de problèmes
- chercher l'information nécessaire
- organiser l'information pour la rendre utilisable
- communiquer avec les autres
- utiliser des outils de conception
- développer sa capacité de création

1.3 Les algorithmes

La phase heuristique conduit à élaborer un jeu d'instructions qui précise exactement comment résoudre un problème particulier en un nombre fini d'étapes. Cette partie de la résolution implique donc l'élaboration d'un algorithme précis, clair et spécifique.

La méthode de travail oblige à rechercher de l'information et à organiser cette information. Cette méthode est l'ensemble des moyens qu'on juge les plus pertinents pour atteindre un but.

Les techniques d'utilisation d'une machine ainsi que les connaissances d'un langage de programmation ne sont que des outils qui permettent de concrétiser le projet qu'on s'est donné. Il est donc primordial de ne pas emprisonner la démarche dans un cadre linguistique.

1.4 Les règles d'actions:

L'apprentissage des schémas informatiques de base permettront à l'étudiant de résoudre un problème précis. Ces structures informatiques sont:

- les structures séquentielles,
- les structures répétitives et
- les branchements conditionnels.

1.5 Les étapes de résolution d'un problème

1.5.1 Problème à résoudre

Travail de la caissière.

J'ai fait des achats pour un montant de 12,77\$. Si je paye avec un billet de 20\$, l'ordinateur indique que la caissière me remettra 1 billet de 5\$, 1 billet de 2\$, 2 pièces de 10¢ et 3 pièces de 1¢.

1.5.2 Résolution:

- 1.5.2.1 Mise en situation et découverte de certains éléments:
remettre des billets et des pièces
résultats anticipés
- 1.5.2.2 Décomposer le problème en sous-problèmes. Ex:
remettre les billets de 5 \$.
- 1.5.2.3 Trouver une stratégie de résolution (algorithme de résolution)
2 choses :
 - nombre entier de fois que 5 est contenu dans le montant à remettre
 - montant qui reste à rendre
- 1.5.2.4 Revenir au problème initial i.e. remettre des billets et des pièces.
- 1.5.2.5 Résoudre les autres problèmes par analogie.
billet 2, billet 1, pièces 50 . . .

1.6 Autres exemples de problèmes à résoudre.

Les problèmes qui suivent comportent une description des grandes étapes de résolution illustrées à l'aide de l'arbre algorithmique.

1.6.1 Dessiner une fusée

- 1.6.1.1 Déterminer les parties du dessin
- 1.6.1.2 Faire l'arbre algorithmique cf. FIG.
- 1.6.1.3 Traduire dans un langage de programmation

1.6.2 Calculer la consommation d'essence d'une auto

- 1.6.2.1 Déterminer les entrées-sorties (Km, litres)
litres/100Km, milles/gallon
- 1.6.2.2 Déterminer l'algorithme de résolution
Calculer la consommation:
 - règle de trois
 - Résultat 1 $\leftarrow$ litres/Km X 100
 - formule de Transport Canada
 - Résultat 2 $\leftarrow$ 282.48/RÉSULTAT 1
- 1.6.2.3 Faire l'arbre algorithmique cf. FIG 2
- 1.6.2.4 Traduire dans un langage de programmation

N.B. D'autres exemples d'arbres algorithmiques sont illustrés dans les pages qui suivent.

1.7 Les apprentissages de base

Les problèmes à résoudre sont multiples et reflètent les préoccupations suivantes:

- afficher des textes sur l'écran
- faire un dessin
- animer un objet
- déplacer un objet
- faire des calculs
- faire de la musique

1.8 La programmation structurée

Le langage de programmation BASIC ne favorise pas toujours ce que l'on appelle la programmation structurée. Il est donc important de bien structurer la démarche préalable à la codification proprement dite.

Les gabarits qui suivent illustrent bien les structures alternatives et les branchements conditionnels.

Si ... Alors ...
Sinon ...

Nécessité d'utiliser GOTO

- ① module VRAI
- ② module FAUX

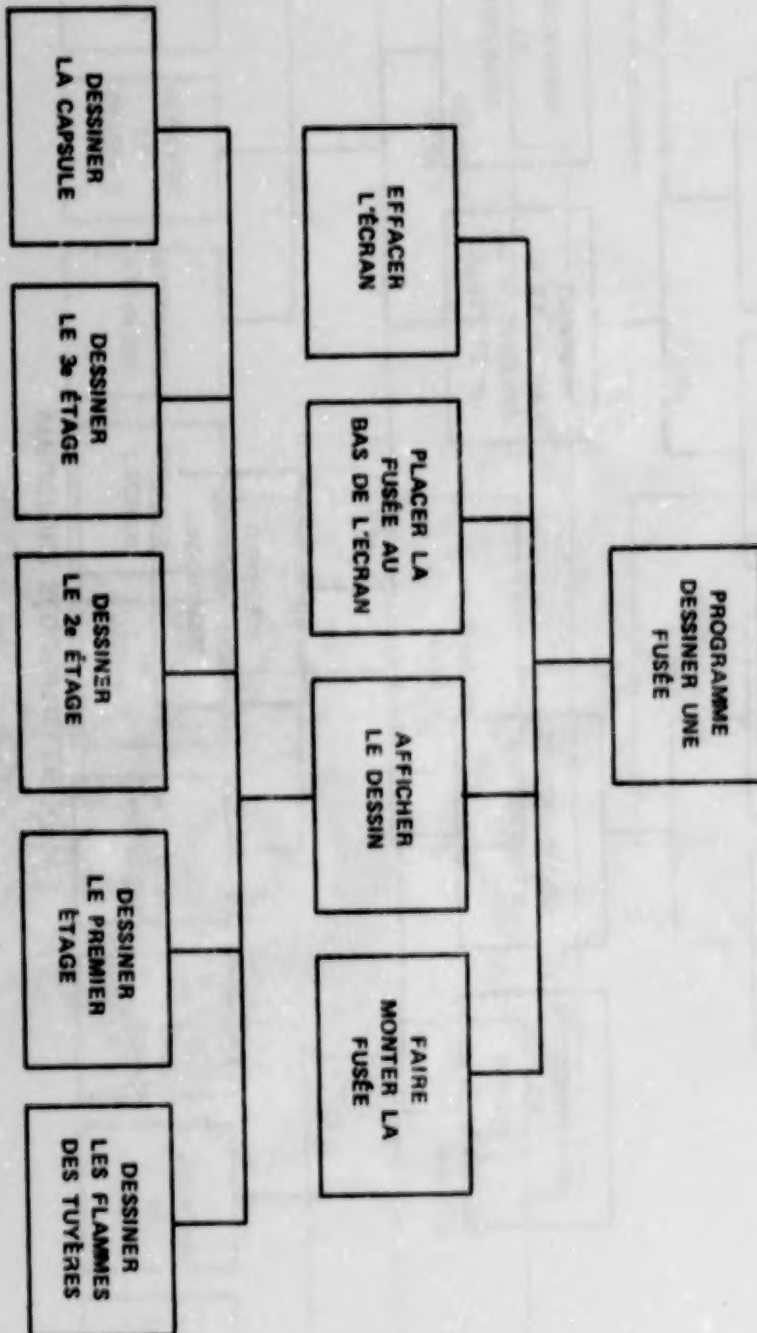
MODULE VRAI

```
100 IF ... THEN 120 'SI
110 GOTO 150
120
130     MODULE VRAI
140
150     'FINSI
```

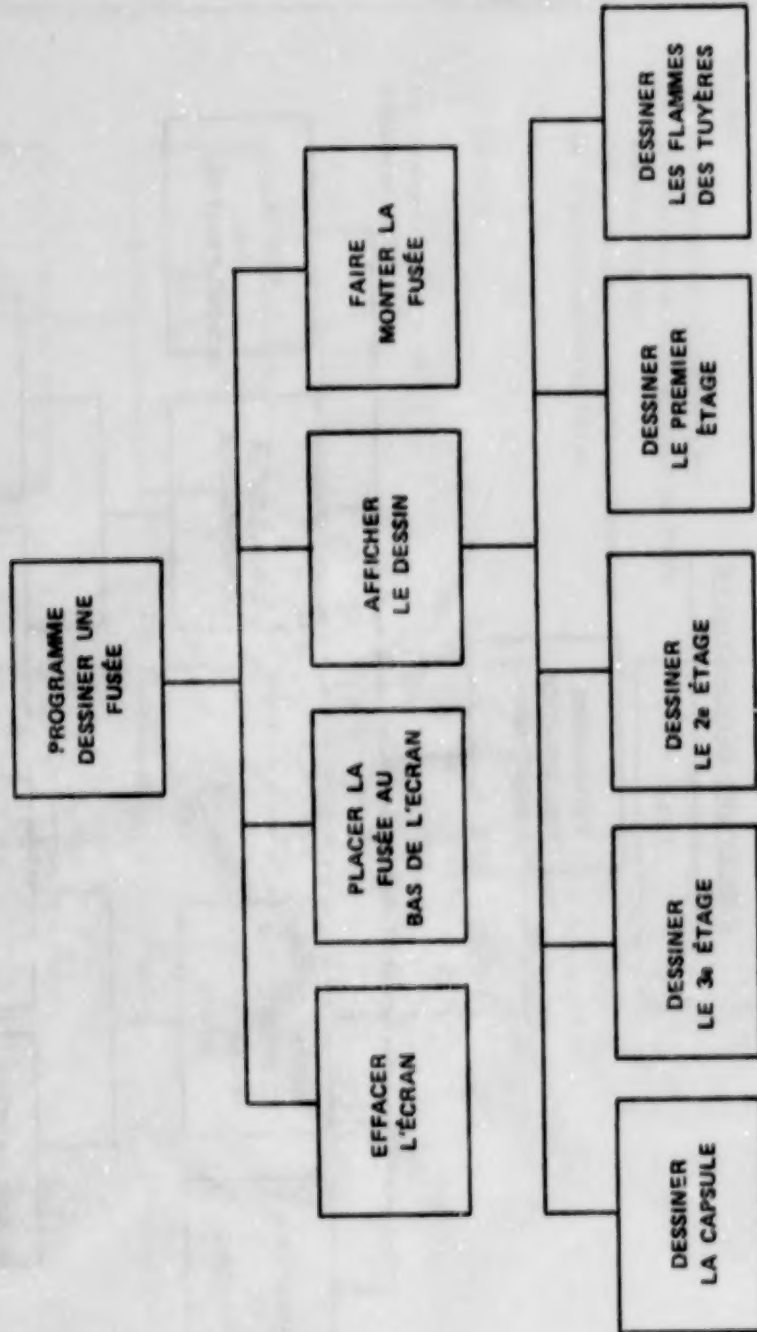
UN MODULE VRAI ET UN FAUX

```
100 IF ... THEN 120 'SI
110 GOTO 160
120
130     MODULE VRAI
140
150 GOTO 200
160     'SINON
170
180     MODULE FAUX
190
200     'FINSI
```

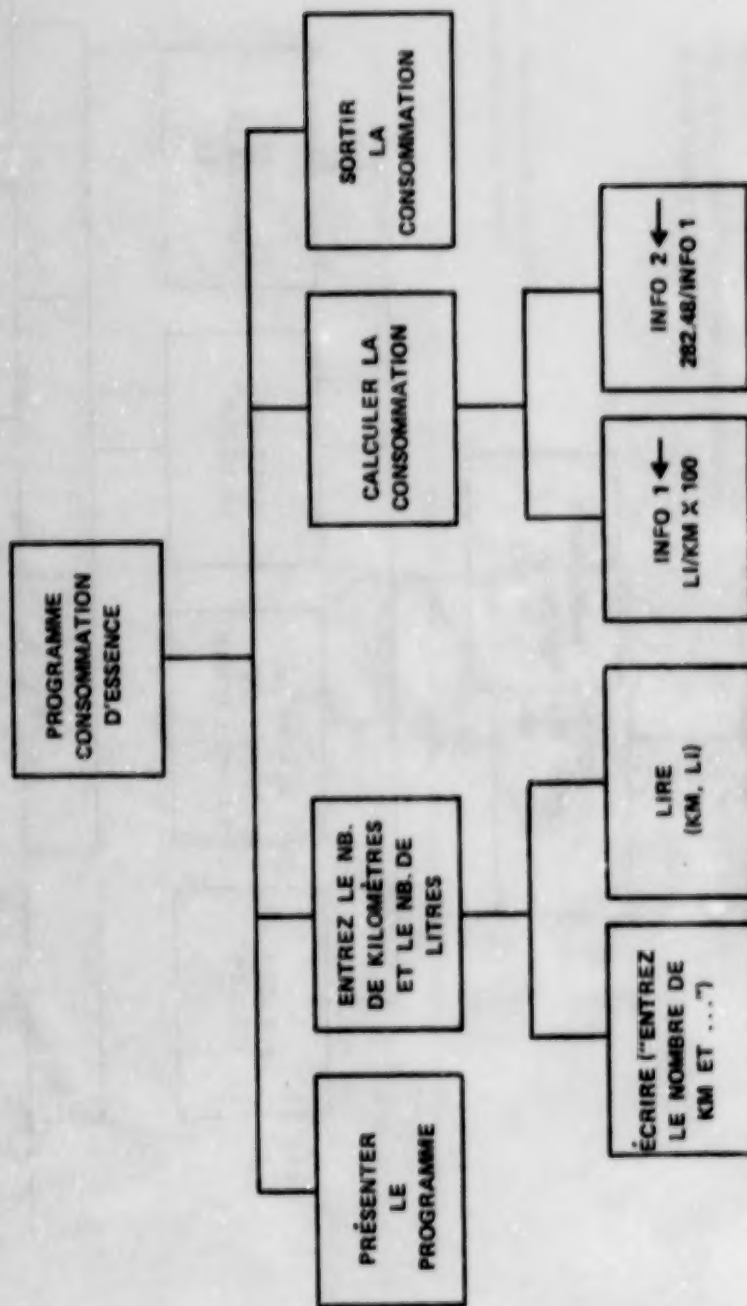
STRUCTURE SÉQUENTIELLE



STRUCTURE SÉQUENTIELLE



STRUCTURE SÉQUENTIELLE



PROGRAMME D'ANIMATION

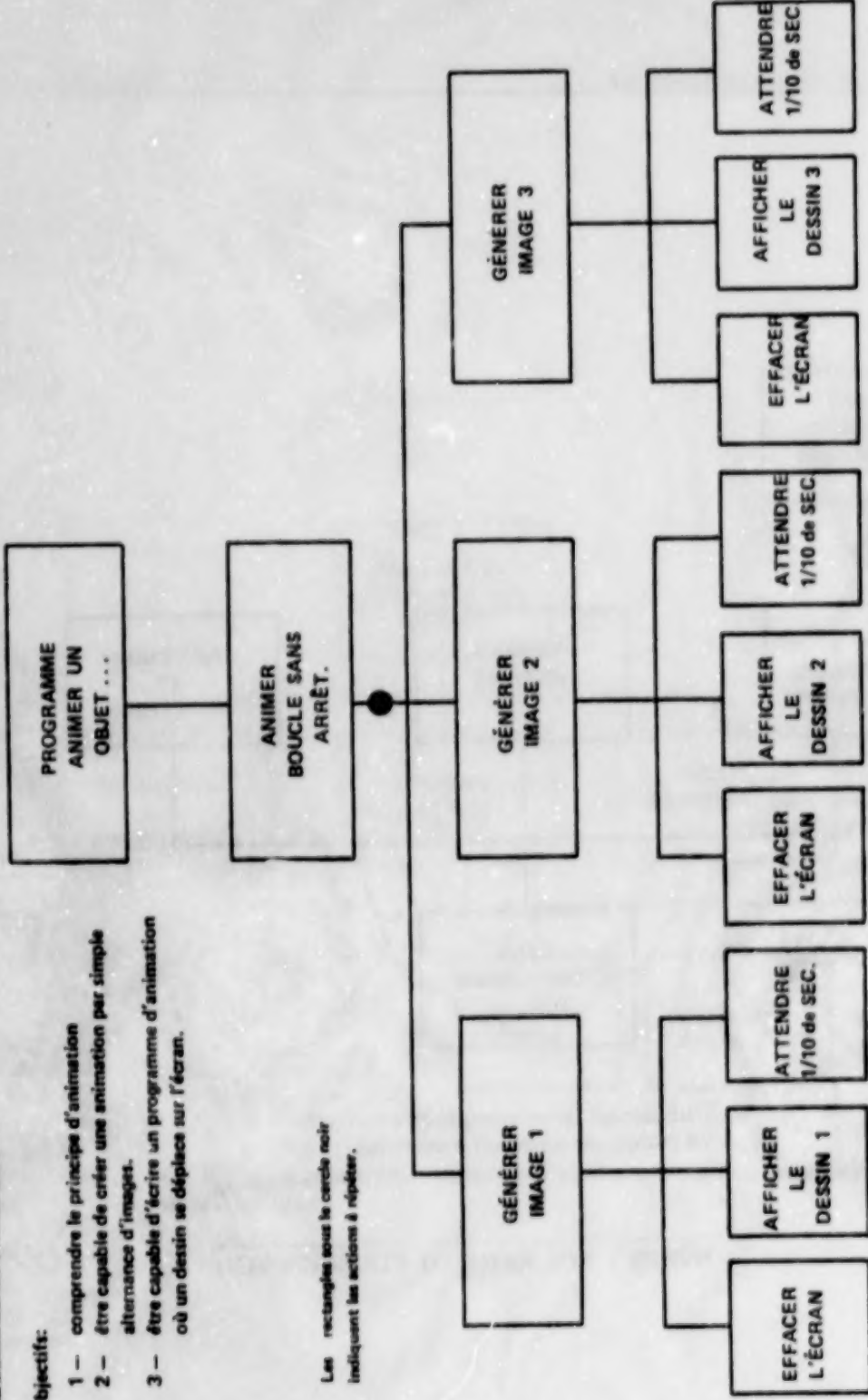
Objectifs:

- 1 - comprendre le principe d'animation
- 2 - être capable de créer une animation par simple alternance d'images.
- 3 - être capable d'écrire un programme d'animation où un dessin se déplace sur l'écran.

Les rectangles sous le cercle noir indiquent les actions à répéter.

PRINCIPE D'ANIMATION

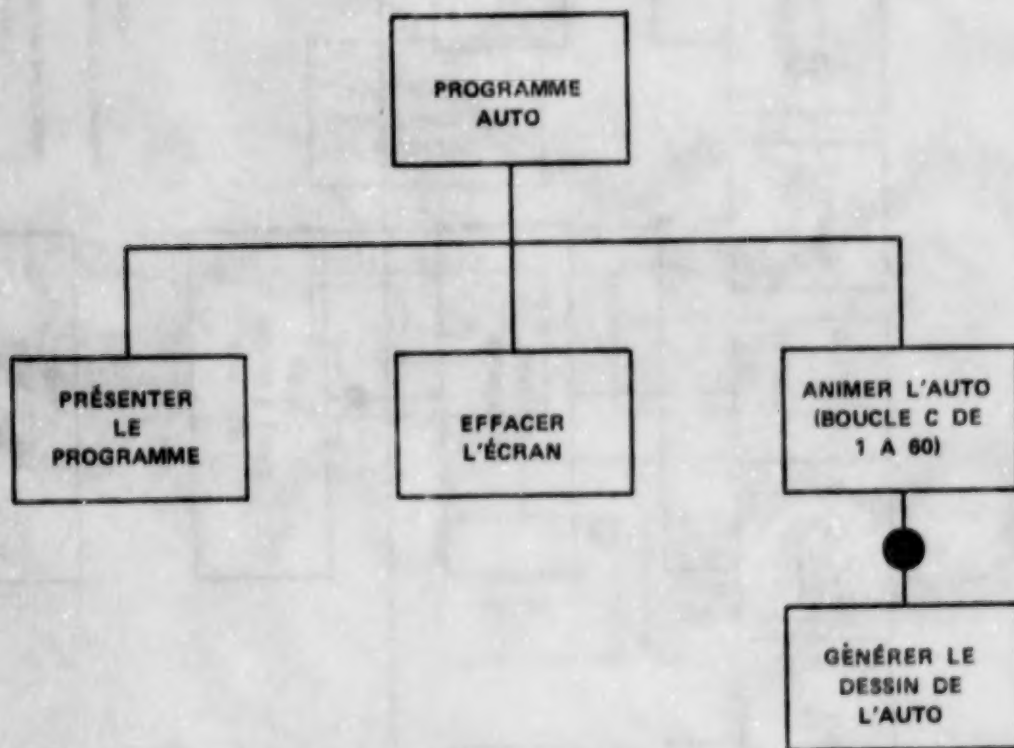
Répétition rapide d'une suite d'images à l'aide de structures répétitives (BOUCLE D'ANIMATION) et de délais (BOUCLE DE TEMPORISATION)

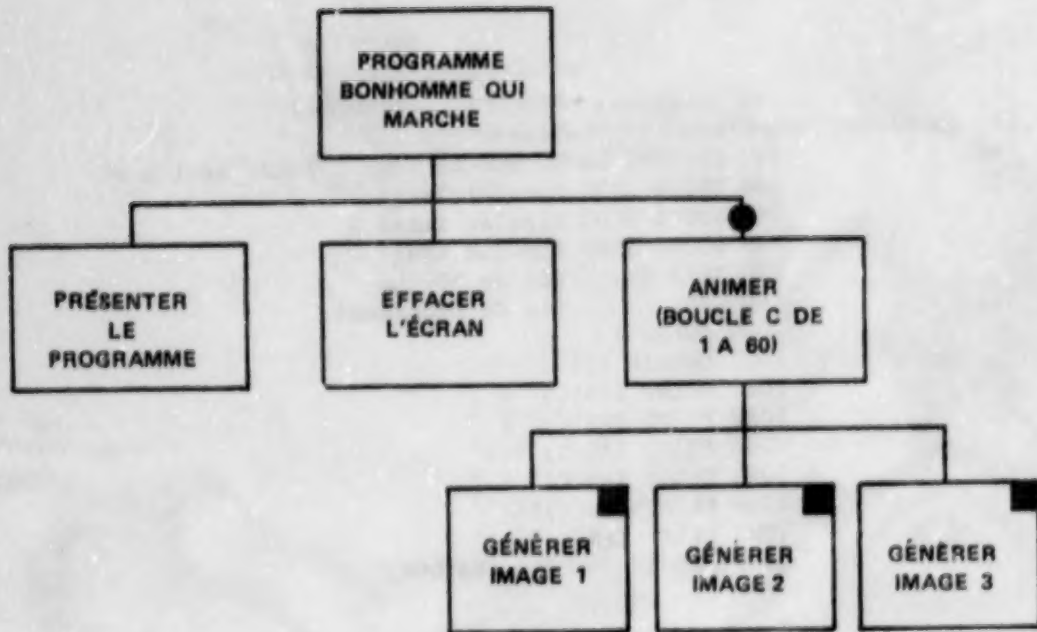


DÉPLACEMENTS D'OBJETS SUR L'ÉCRAN

Principe :

Translation d'un dessin sur l'écran au moyen de la fonction de tabulation.





N.B. : Le ■ dans le coin droit supérieur d'une boîte indique un sous-programme ou une procédure.

Exemples de programmes en BASIC illustrant
les arbres algorithmiques décrits précédemment

```

90 programme AUTO
100 cls      effacer l'écran
110 for C=1 to 60 BOUCLE dont C passe de 1 à 60
120 LOCATE 1,1
130 PRINT TAB(C)" : ==/ "
140 PRINT TAB(C)" /----- "
150 PRINT TAB(C)" / -0---0-- "
160 NEXT C    fin de BOUCLE
170 cls      effacer l'écran
180 end      Fin
  
```

```

90 programme BONHOMME QUI MARCHE
100 cls      effacer l'écran
110 for C=1 to 60 BOUCLE dont C passe de 1 à 60
120 GOSUB 1000 Appeler image 1
130 GOSUB 2000 Appeler image 2
140 GOSUB 3000 Appeler image 3
150 NEXT C    fin de BOUCLE
160 END      fin du programme

1000 image 1
1005 LOCATE 1,1
1010 PRINT TAB(C)" 0 "
1020 PRINT TAB(C)" 1 "
1030 PRINT TAB(C)"-1-"
1040 PRINT TAB(C)" 1 "
1050 PRINT TAB(C)"/ "
1060 PRINT TAB(C)"/ 1"
1999 RETURN    fin (RETOUR)
  
```

IMAGE 3

```

0
1
-1-
1
/
  
```

IMAGE 2

```

0
1
/1-
1
/
/
  
```

FIG. 7

1.9 Les méthodes d'analyse

Certaines méthodes d'analyse, comme la décomposition d'un problème en sous-problèmes, sont souvent utilisées tandis que d'autres le sont moins. Nous définissons brièvement ces stratégies de résolution qui, dans certains cas, sont spécifiques au 5e secondaire.

Ces méthodes d'analyse sont:

- l'inférence (4,5)
- la méthode progressive (4)
- la méthode régressive (5)
- la décomposition d'un problème en sous-problèmes (4)
- la méthode d'essais et d'erreurs (4)
- la contradiction (5)

1.10 La méthode de résolution et les stratégies de résolution

1.10.1 La méthode heuristique comprend:

1.10.1.1 Une suite d'essais donnant des résultats approchés

1.10.1.2 Un contrôle de la progression vers un résultat final acceptable

1.10.2 L'apprentissage de la méthode heuristique s'acquiert par:

1.10.2.1 L'expérience personnelle de résolution de problèmes

1.10.2.2 L'observation de gens qui résolvent des problèmes

1.10.3 Les stratégies de résolution de problèmes sont:

1.10.3.1 L'inférence (induction et déduction)

1.10.3.2 La méthode régressive

1.10.3.3 La décomposition d'un problème en sous-problèmes

1.10.3.4 La méthode d'essais et d'erreurs

1.10.3.5 La contradiction

1.11 Mise en situation

Selon les préférences et les goûts personnels des étudiants, des problèmes spécifiques sont apportés.

L'enseignant intervient par des questions qui incitent l'étudiant à utiliser des éléments de méthode pour la résolution de problèmes (des heuristiques); il favorise une certaine incertitude critique en encourageant des approches multiples aux solutions.

Pour l'étudiant, il ne suffit pas de deviner la solution que le professeur a en tête. Au contraire, il s'agit de bâtir sa propre solution, le professeur pouvant aider sur le plan de la méthode. Les activités de résolution de problèmes se situent dans un contexte d'enseignement heuristique.

Résoudre des problèmes adaptés au niveau de connaissance des étudiants est plus riche que faire de petits exercices d'application qu'on demande habituellement aux étudiants. Il faut, pour l'étudiant, une activité créatrice.

1.11.1 La résolution de problèmes a pour buts:

- 1.11.1.1 De développer l'habileté et la confiance en soi devant un problème nouveau
- 1.11.1.2 De développer un algorithme de résolution
- 1.11.1.3 D'être capable de construire une solution à partir de ses propres erreurs
- 1.11.1.4 De valoriser l'activité de l'étudiant

1.11.2 Liste de problèmes à résoudre:

- faire cuire un gâteau
- faire la lessive
- assembler une auto modèle réduit
- organiser une joute de hockey
- préparer quelque chose à manger
- participer à la formation I.S.I.
- faire un voyage
- faire une recherche en géographie
- faire une partie de cartes
- mettre le microordinateur en fonction et coder mon projet
- préparer mon équipement de sport pour la saison qui vient
- réaliser un jeu de déduction genre "Mastermind"
- traduire une partition musicale
- simuler un appel téléphonique
- faire un questionnaire assisté par ordinateur
- remplir le constat amiable à la suite d'un accident
- faire un poème
- etc.

1.11.3 Tâches à accomplir

- trouver les éléments d'un problème
- trouver des solutions partielles et/ou globales
- développer des algorithmes

1.12 L'inférence

1.12.1 Définition

L'inférence est un outil heuristique de résolution de problèmes. C'est l'opération intellectuelle par laquelle on passe d'une vérité à une autre. Nous concluons à une proposition vraie à partir de propositions prises comme hypothèses (processus de déduction).

Nous généralisons une observation ou un raisonnement à partir de cas singuliers et dégageons les informations explicites et implicites (processus d'induction).

1.12.2 PROBLÈME à résoudre:

Des poules et des cochons

Deux enfants visitent une ferme. De retour à la maison, le garçon raconte qu'il a vu 18 animaux tandis que sa soeur a compté 50 pattes. Combien y avait-il de poules et de cochons?

1.12.3 RÉOLUTION du problème:

Nous cherchons le nombre de poules et de cochons. Par expérience nous savons qu'une poule a deux(2) pattes et un cochon en a quatre(4).

1.12.4 Établir les relations qui existent entre les éléments.

Nombre de pattes:

Le nombre de poules $\times 2 =$

le nombre de cochons $\times 4 = 50$

$$\text{Cochons} + \text{Poules} = 18$$

Si 1 cochon alors 17 poules
nous avons $4 \times 1 \text{ cochon} + 2 \times 17 \text{ poules} = 38 \text{ pattes}$

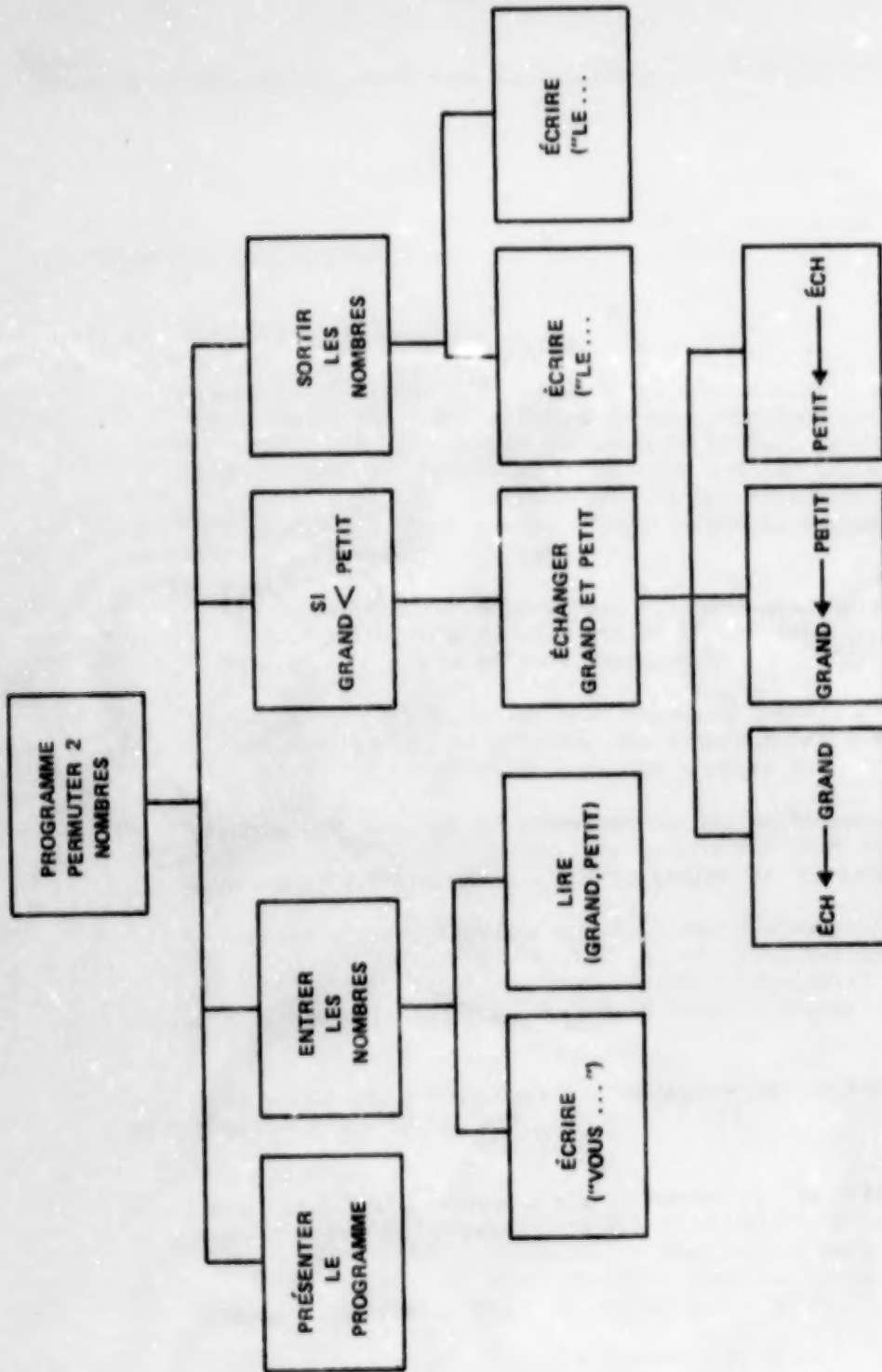
Si 2 cochons alors 16 poules
nous avons $4 \times 2 \text{ cochons} + 2 \times 16 \text{ poules} = 40 \text{ pattes}$

... et ainsi de suite pour en arriver à ...

7 cochons alors 11 poules

NOUS AVONS $4 \times 7 \text{ COCHONS} + 2 \times 11 \text{ POULES} = 50 \text{ PATTES}$

RÉPONSE : 7 cochons et 11 poules



1.13 La méthode régressive

1.13.1 Historique et description

Papus d'Alexandrie (3e siècle av. J.-C.) a décrit et commenté les travaux des mathématiciens grecs dans une série de huit volumes. Les écrits de Papus devinrent, après les travaux d'Euclide, les ouvrages les plus lus par les mathématiciens de l'époque. Le livre VII est une recension des découvertes des géomètres grecs. Papus décrit la méthode d'analyse et de synthèse utilisée.

C'est aussi loin que l'on puisse remonter dans le temps, la première description de résolution de problèmes connue de nos jours sous le nom de méthode régressive.

L'analyse permet de partir d'un résultat obtenu, d'en chercher l'antécédent, de retracer les étapes antérieures pour en arriver à quelque chose de connu et d'accepté.

La synthèse ordonne les conséquences qui antérieurement étaient, les interrelie les unes aux autres pour en arriver à la solution recherchée.

Une solution à un problème comporte des étapes qui conduisent de l'information donnée au but. Il est plus facile de partir du but pour en arriver à l'état initial (analyse). Ceci étant accompli, la solution est la série d'étapes dans l'ordre inverse.

1.13.2 PROBLÈME à résoudre:

Pile ou face

Trois étudiants lancent une pièce de monnaie! Celui qui obtient une figure différente des deux autres perd et doit doubler leur avoir. Après trois(3) parties, chaque joueur a perdu une fois et l'avoir de chacun est de 24\$.

Combien d'argent avait chaque étudiant au début?

1.13.3 Étapes de RÉSOLUTION du problème:

1.7.3.1 Élaborer un plan

1.7.3.2 Exécuter le plan

1.7.3.3 Vérifier les résultats

1.13.4 Résolution du problème

Soient les étudiants A, B et C.

Si C a perdu la troisième partie, alors A et B avaient en leur possession 12\$ et C avait 48\$.

Si B a perdu la deuxième partie, alors A avait 6\$ et C avait 24\$. B avait donc 42\$.

C'est ainsi que nous retrouvons l'avoir de chacun au début de la partie.

1.13.5 Tableau récapitulatif de la situation.

	A	B	C
3	24	24	24
2	12	12	48
1	6	42	24
Début	39	21	12

1.14 La décomposition d'un problème en sous-problèmes

1.14.1 Définition

La méthode consiste à:

1.14.1.1 Résoudre chacune des parties

1.14.1.2 Regrouper les parties pour résoudre le problème initial.

Problèmes -----> sous-problèmes -----> sous-problèmes

N.B. Bien souvent nous utilisons cette méthode sans trop le savoir, celle-ci étant tellement évidente.

1.14.2 PROBLÈME à résoudre:

1.14.2.1 Problème: Façon de rendre la monnaie.
Combien y a-t-il de façons de rendre la monnaie de 20¢ en pièces de 1¢, 5¢ et 10¢.

1.14.2.2 Résolution: La résolution se fera étape par étape.

Si nous utilisons seulement des 1¢, il y a une seule façon de rendre la monnaie.

Si nous utilisons des 5¢ et des 1¢, nous donnons un nombre de 5¢ soit 0, 1, 2, 3 ou 4 et le reste en 1¢.

À chaque étape, nous réduisons le problème à un problème antérieurement résolu, un problème plus simple.

Si nous connaissons le nombre de 5¢ et 1¢ il est facile d'étendre au nombre de 10¢.

1.14.2.3 Résolution: À partir de l'information donnée, il est évident que plusieurs étapes seront nécessaires pour résoudre le problème!

Ce processus permet de se rapprocher progressivement de la solution recherchée en répétant une action qui dépend, en général, du nombre de répétitions déjà effectuées.

Cette partie de programme, qui fait un retour sur le début d'un segment antérieurement parcouru, est une boucle. La boucle n'est intéressante que dans le mesure où elle est évolutive. Ceci implique qu'un paramètre évolue entre une valeur initiale et une valeur finale, à chaque cycle.

La répétition d'une partie d'un programme s'appelle récurrence. Ce traitement exige la connaissance du stade précédent pour effectuer le stade suivant. Comme nous le soulignons, le problème de rendre la monnaie est un processus récurrent. En effet, à chaque nouvelle étape, j'utilise des résultats antérieurement connus pour poursuivre la résolution.

La récurrence peut être fixe ou variable:

- fixe, si elle est incrémentée par l'énoncé du problème ou si elle est incrémentée par l'évolution nécessaire d'une de ses données.
- variable, si elle est déterminée par le traitement même de la fonction récurrente.

Revenons à notre problème de rendre la monnaie et complétons sa résolution.

Total	1¢	1¢ et 5¢	1¢, 5¢ et 10¢
0	1	1	1
5	1	2	2
10	1	3	4
15	1	4	6
20	1	5	9

1.15.3 PROBLÈME à résoudre:

Les taxes

Plusieurs personnes pensent que les riches devraient être plus taxés que les autres. Toutefois certaines politiques s'avèrent plus ou moins efficaces et justes quand elle sont poussées à l'extrême. Ainsi la nouvelle politique d'impôt prévoit que le taux de taxation sera le même que le nombre de milliers de dollars qu'une personne gagne. Par exemple une personne gagnant 5 000 \$ est taxée à 5 % et une autre gagnant 65 000 \$ le sera à 65 %.

La question est la suivante: Quel est le salaire annuel qui, après les déductions d'impôts, donne un résultat maximum?

1.15.4 Stratégie de résolution:

1.15.4.1 Choisir une série de valeurs

1.15.4.2 Calculer les résultats (salaire net)

1.15.4.3 Tester pour dégager la valeur qui laisse la valeur maximale

1.15.5 Résolution

40 000 \$	X	40/100	=	16 000 \$	;	24 000 \$
50 000 \$	X	50/100	=	25 000 \$	;	25 000 \$
60 000 \$	X	60/100	=	24 000 \$	;	24 000 \$

N.B. Il faut observer les résultats de chaque essai pour guider la sélection de nouvelles valeurs pour le prochain essai. Nous avons ainsi une approche plus précise de la solution.

1.16 La contradiction ou la preuve par l'absurde

1.16.1 Définition

La méthode consiste à faire des inférences logiques qui conduisent à une contradiction.

1.16.2 PROBLÈME à résoudre

1.16.2.1 Problème: Deux dindes pèsent ensemble 10 kg.
La petite dinde se vend 50¢ de plus le kg que la grosse.
La petite a coûté 6\$ et la grosse en a coûté 18\$.
Quel est le poids de chacune?

1.16.2.2 Hypothèse 1: La petite pèse 4 kg.

Renseignements obtenus:

- la grosse pèse $10 - 4 = 6$ kg
- la grosse a coûté 18\$ donc $18/6 = 3\$/\text{kg}$
- la petite coûte 50¢ de plus le kg soit $3.50\$/\text{kg}$
- la petite a coûté 6\$ donc $6/3.50\$ = 1.7$ kg

Il y a contradiction!

Après quelque essais nous obtenons la solution:

1.16.2.3 Hypothèse 2: La petite pèse 2 kg.

Renseignements obtenus:

- la grosse pèse $10 - 2 = 8$ kg
- la grosse a coûté 18 \$ donc $18/8 = 2.50/\text{kg}$
- la petite coûte 50¢ de plus le kg soit $3\$/\text{kg}$
- la petite a coûté 6\$ donc $6/3 = 2$ kg

1.17 Les outils de conception

- 1.17.1 Le structurogramme
- 1.17.2 L'ordinogramme
- 1.17.3 Le pseudo-langage
- 1.17.4 Le diagramme de Warnier-Orr
(Jean-Dominique Warnier et Kenneth T.Orr)
- 1.17.5 L'arbre algorithmique

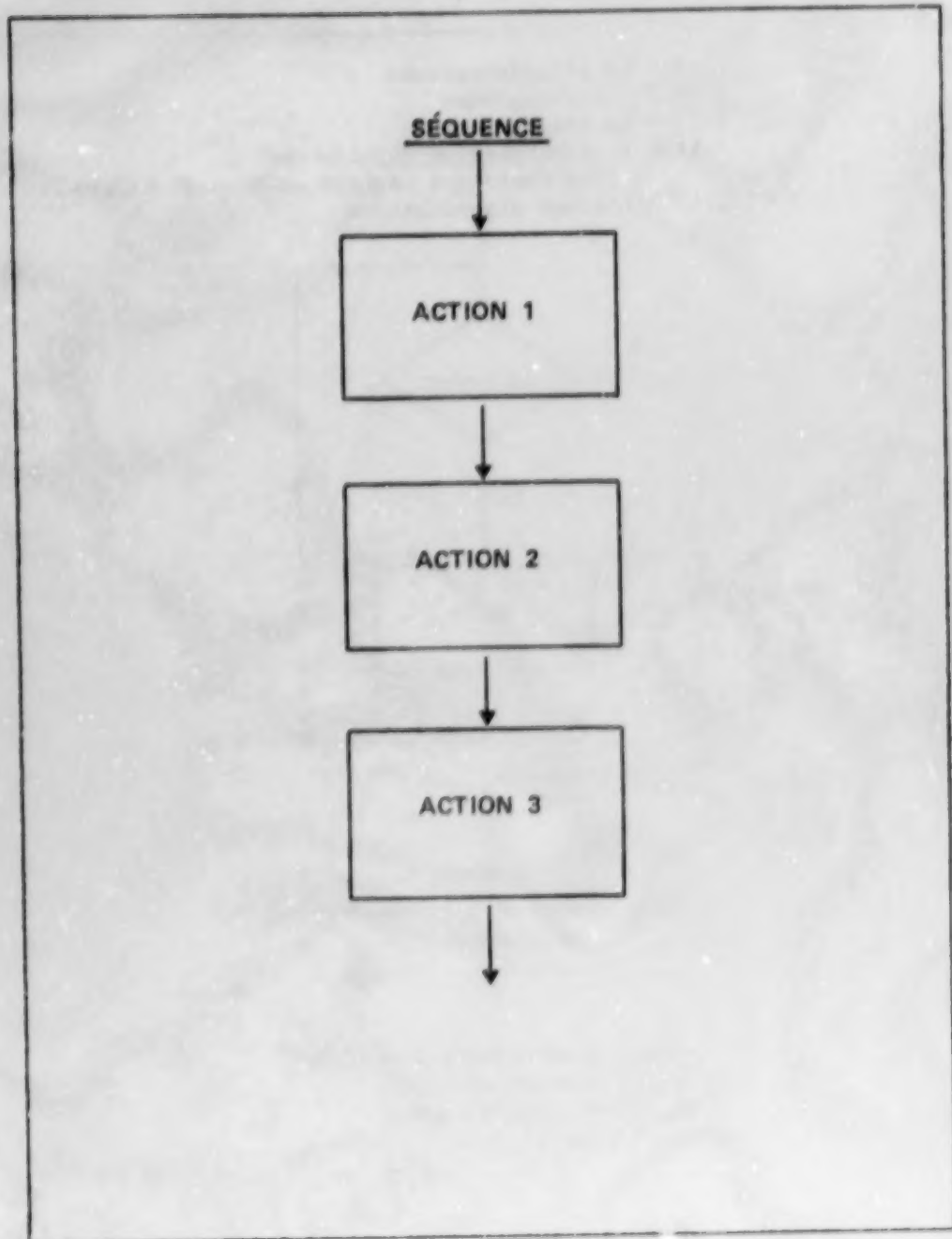


FIG. 8

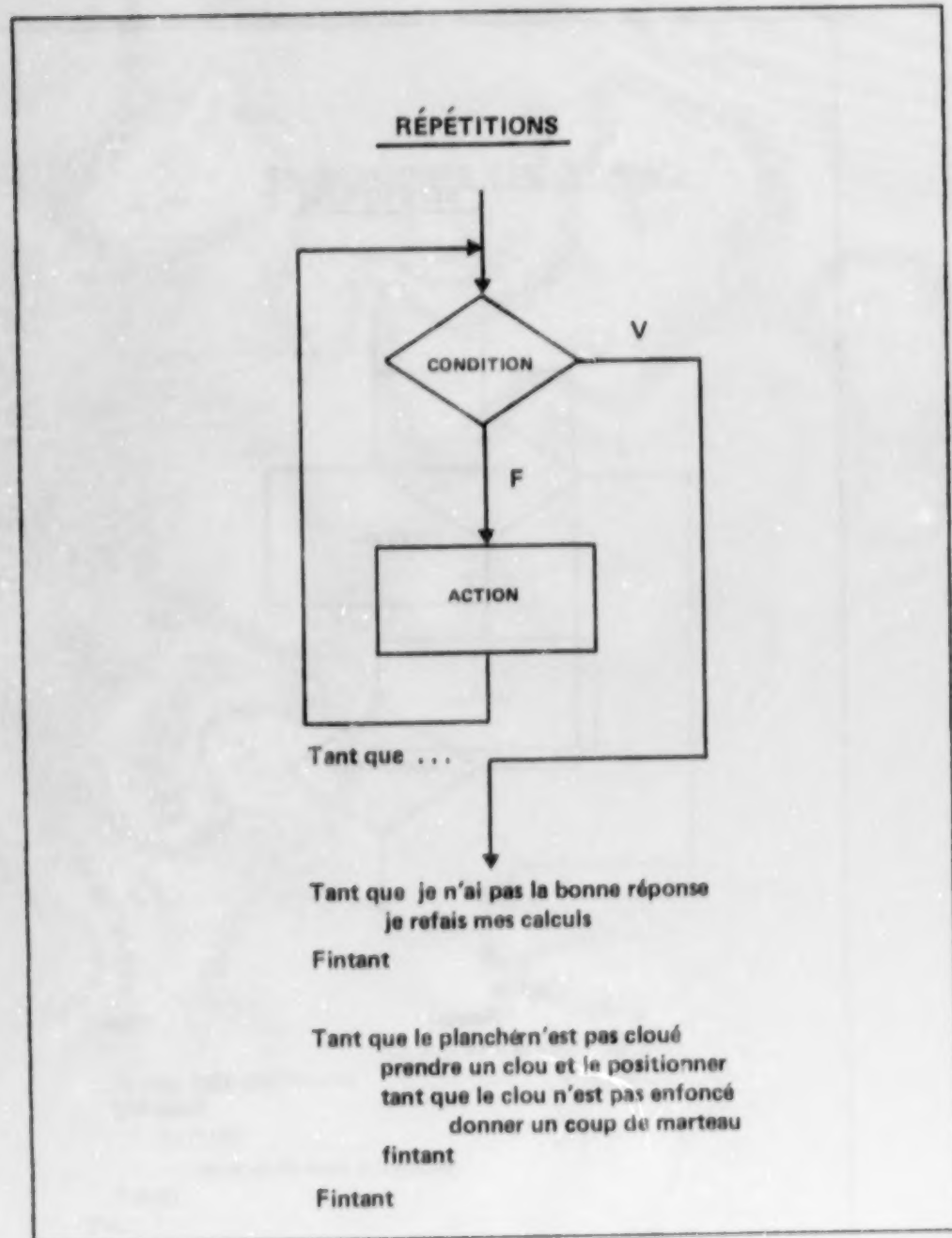


FIG. 9

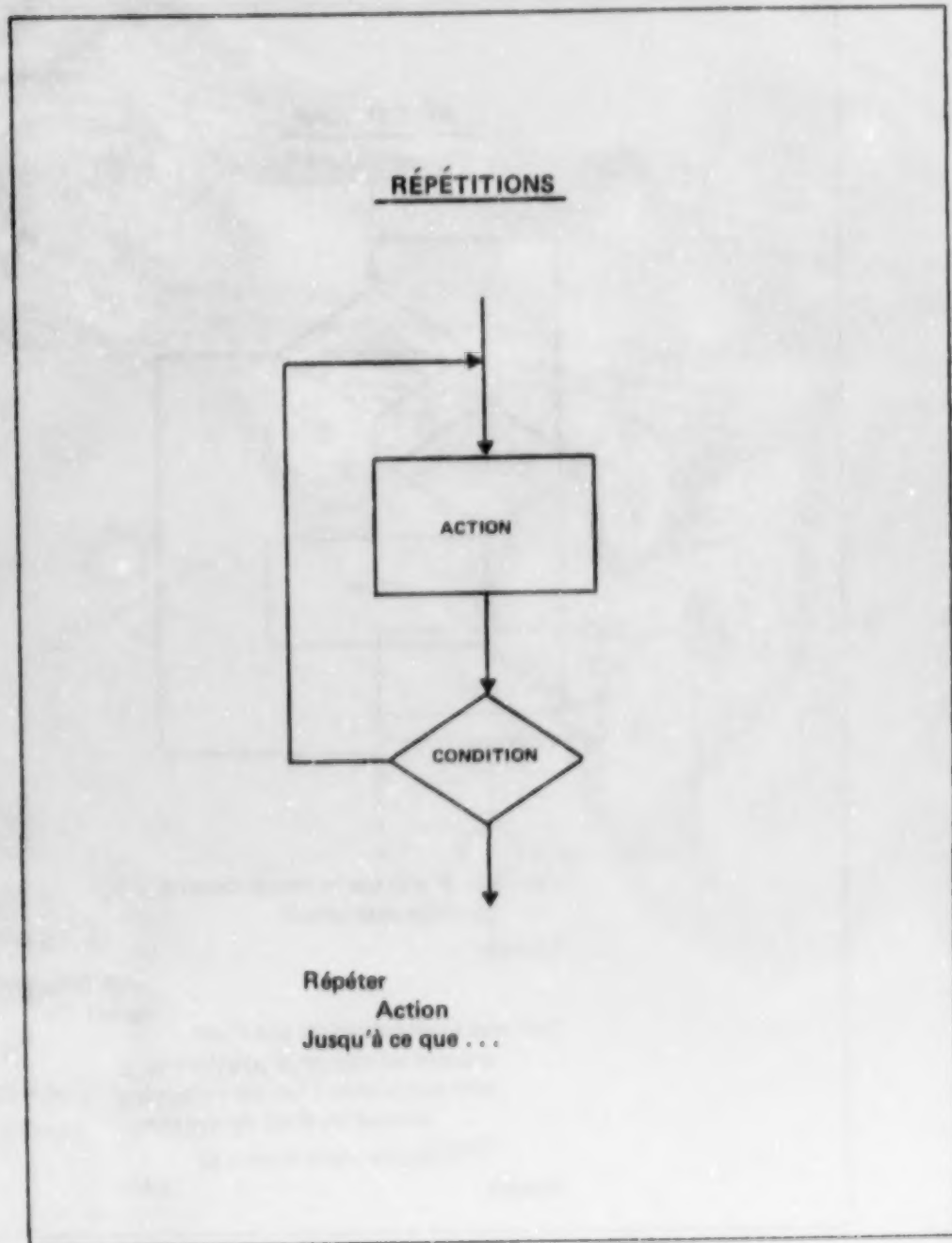
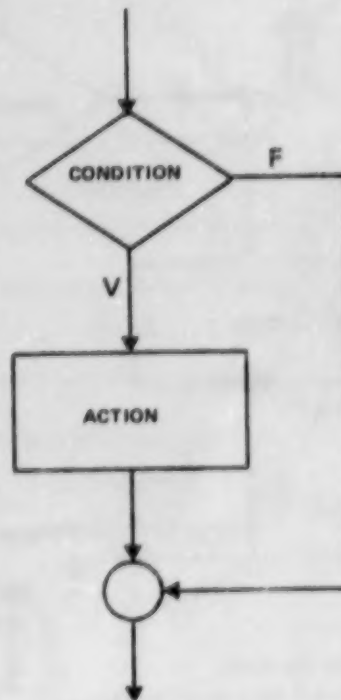


FIG. 10

BRANCHEMENTS CONDITIONNELS



Début

Je vais faire une course

S'il pleut

ALORS

Je prends mon parapluie

FINSI

Fin

FIG. 11

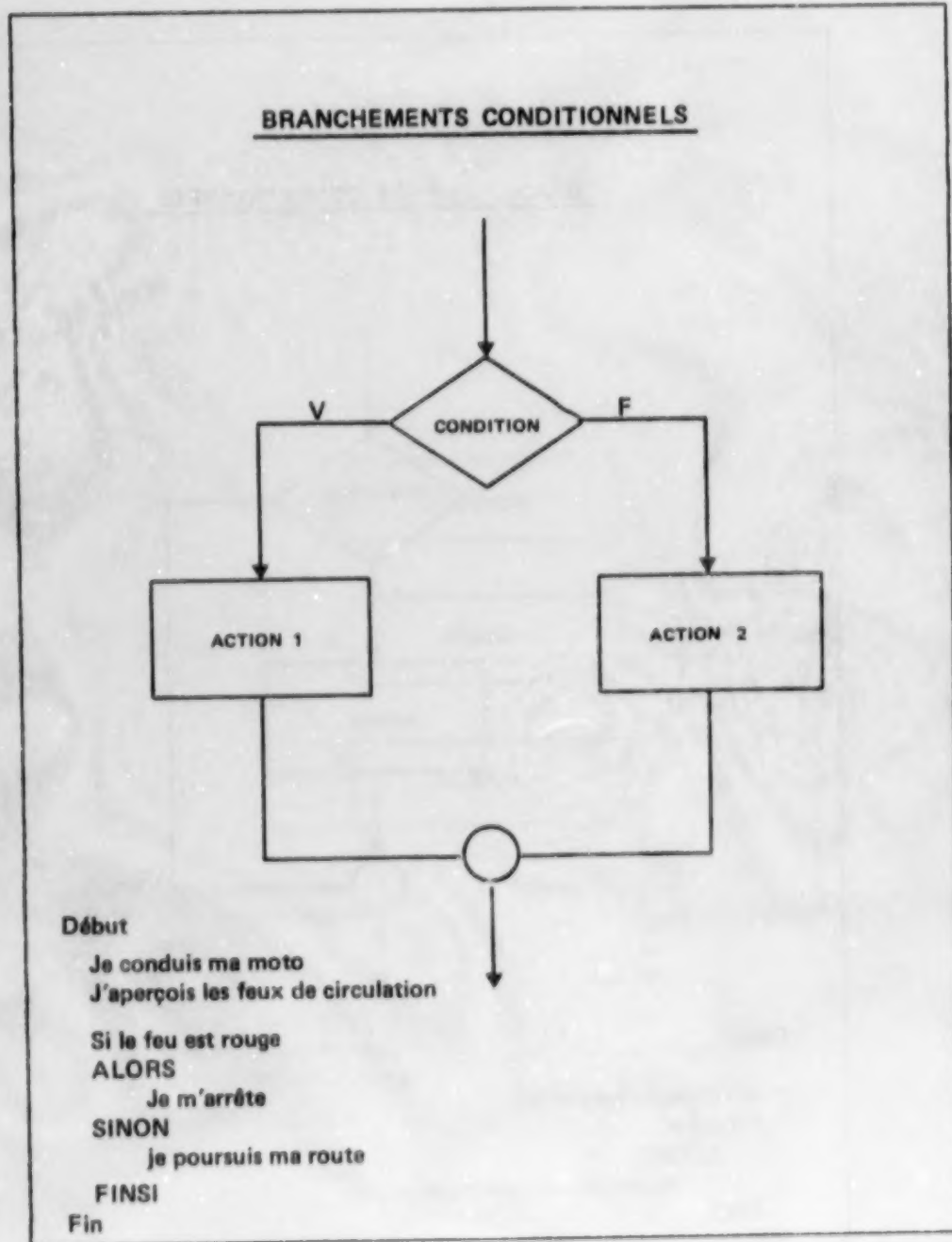


FIG. 12

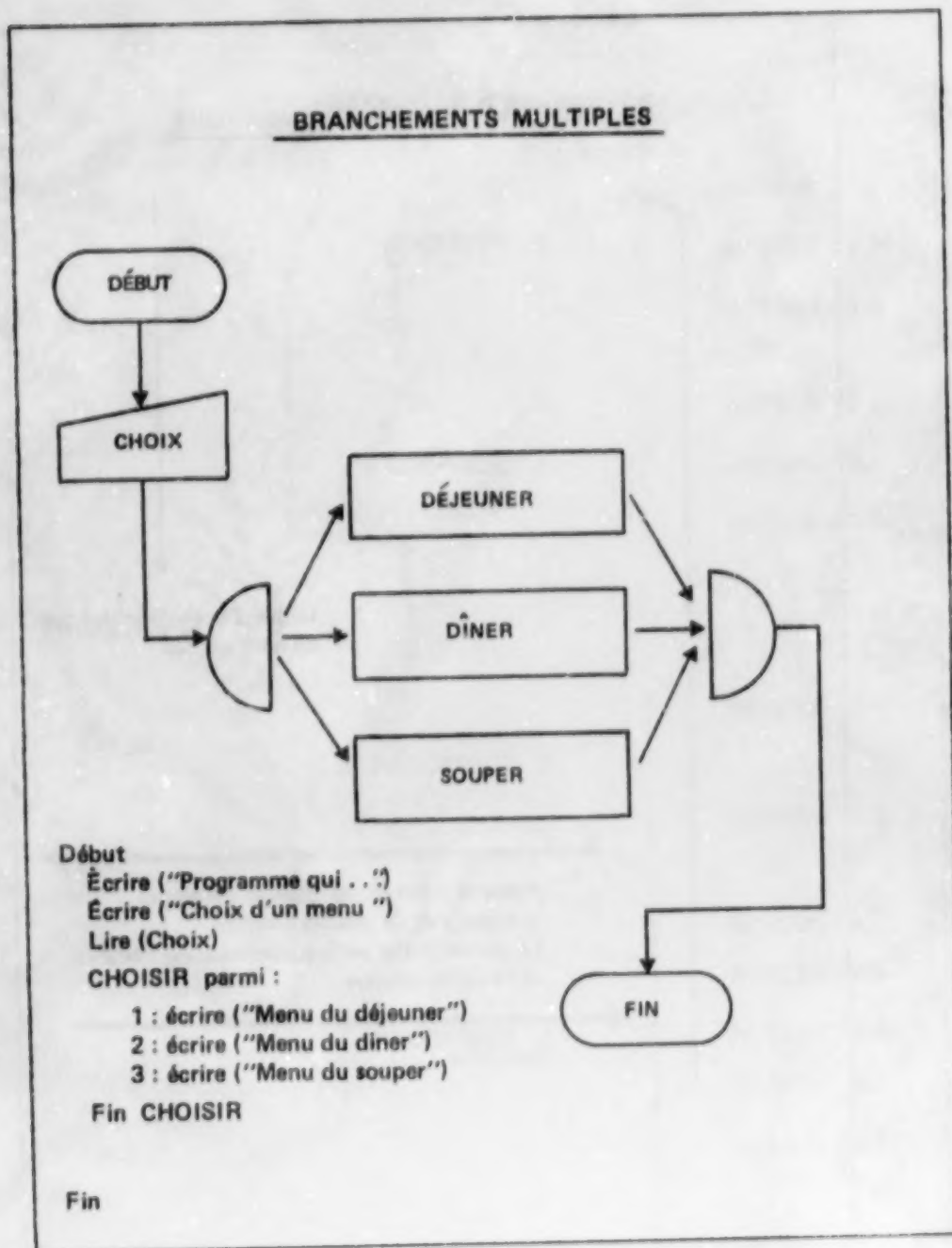


FIG. 13

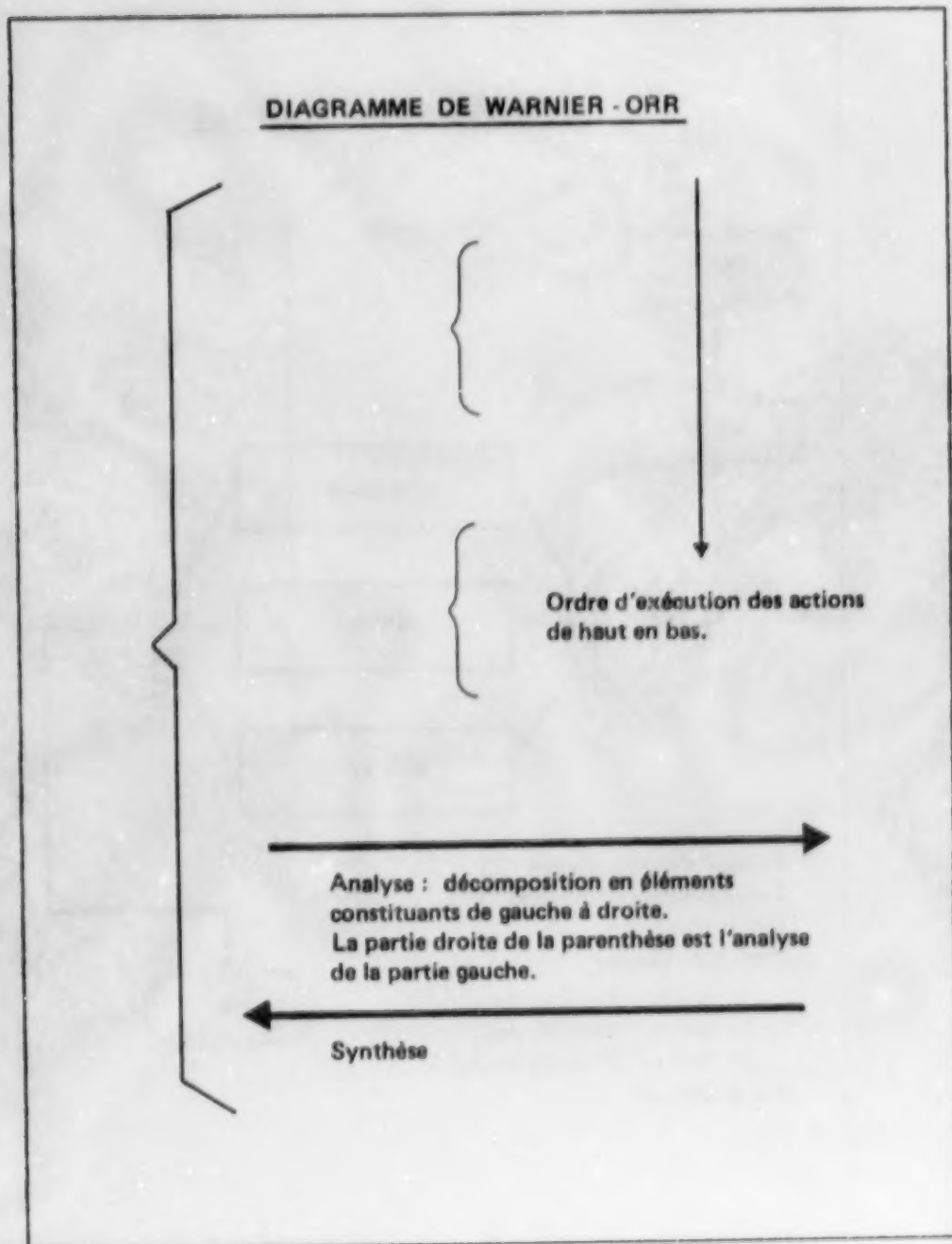


FIG. 14

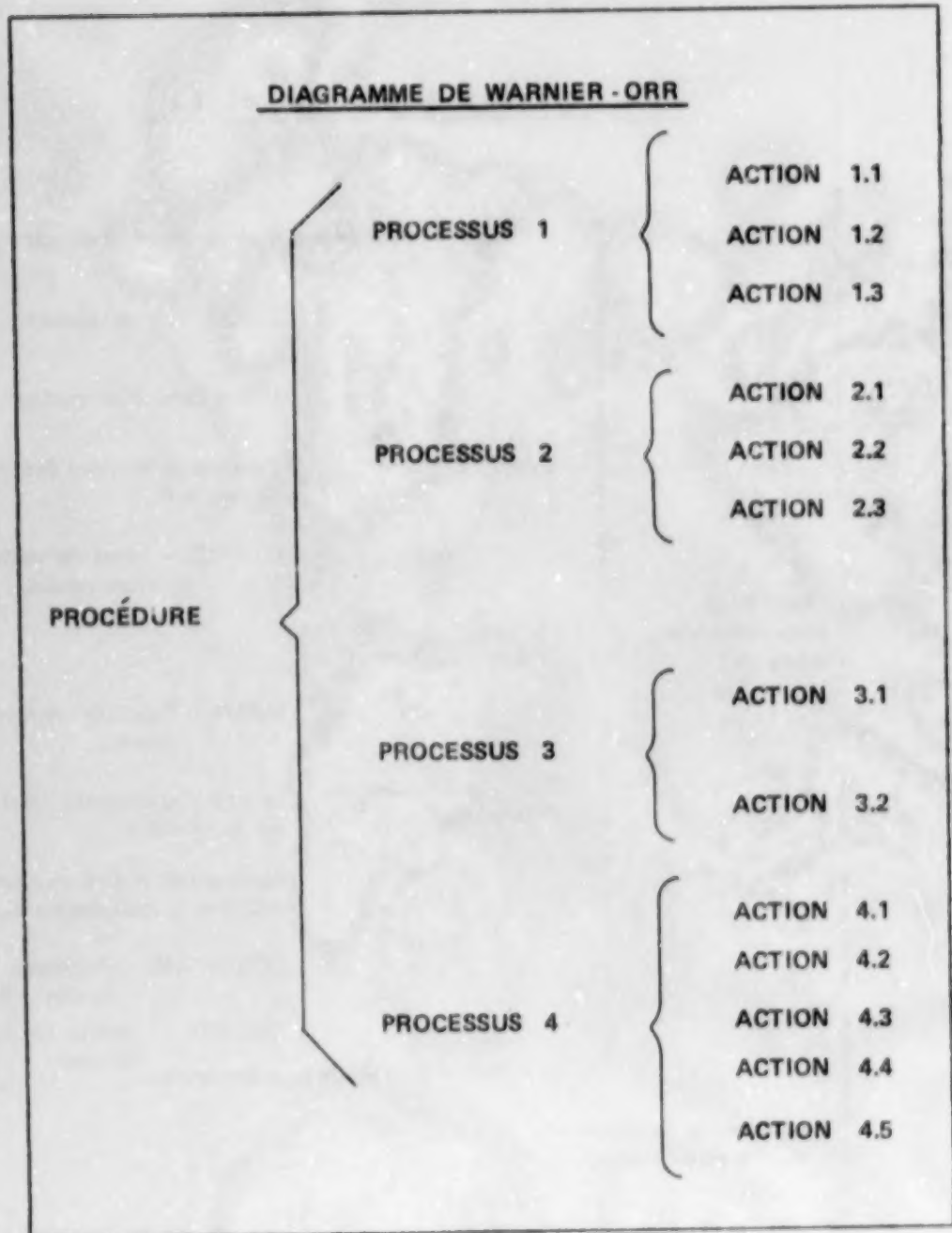


FIG. 15

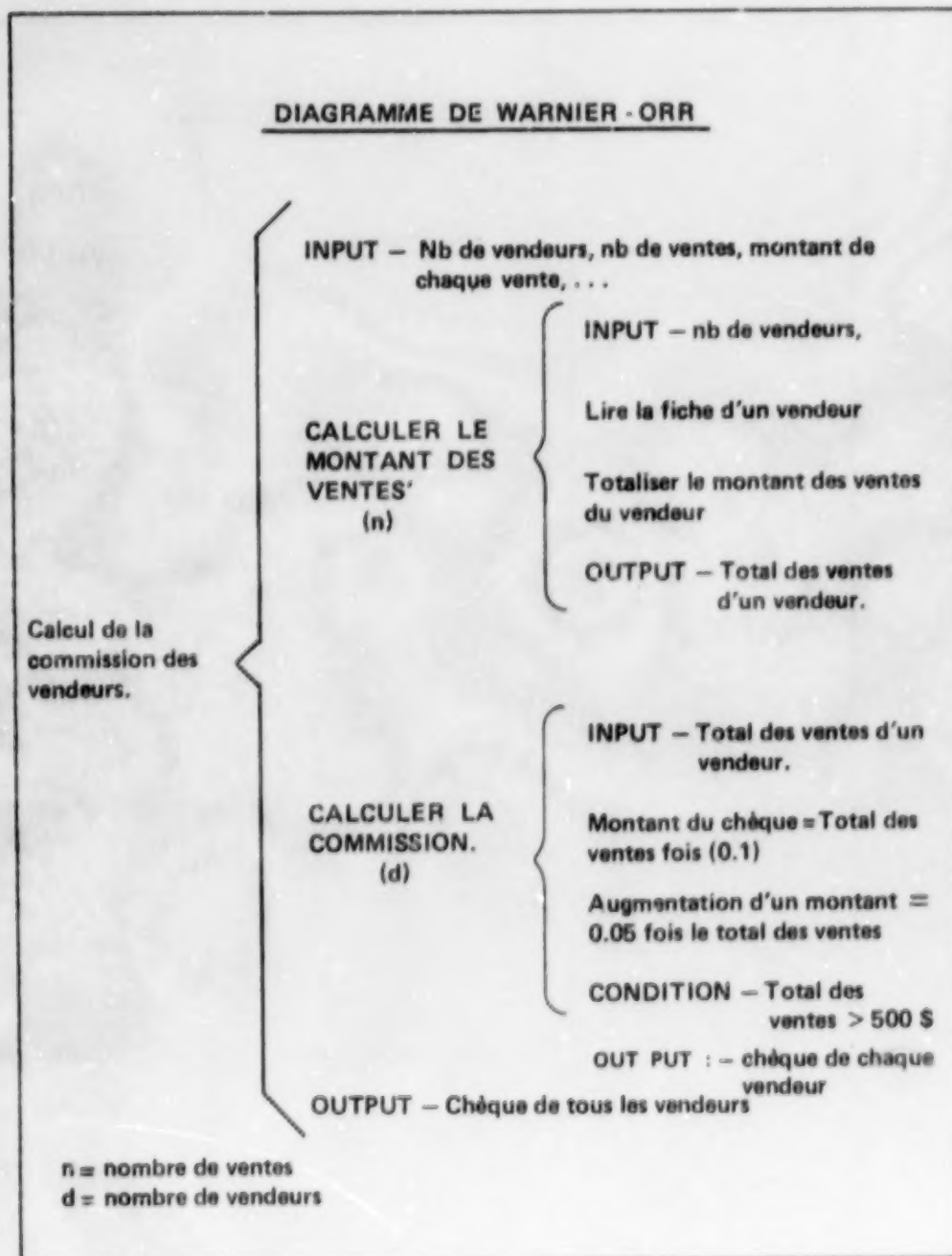


FIG. 15

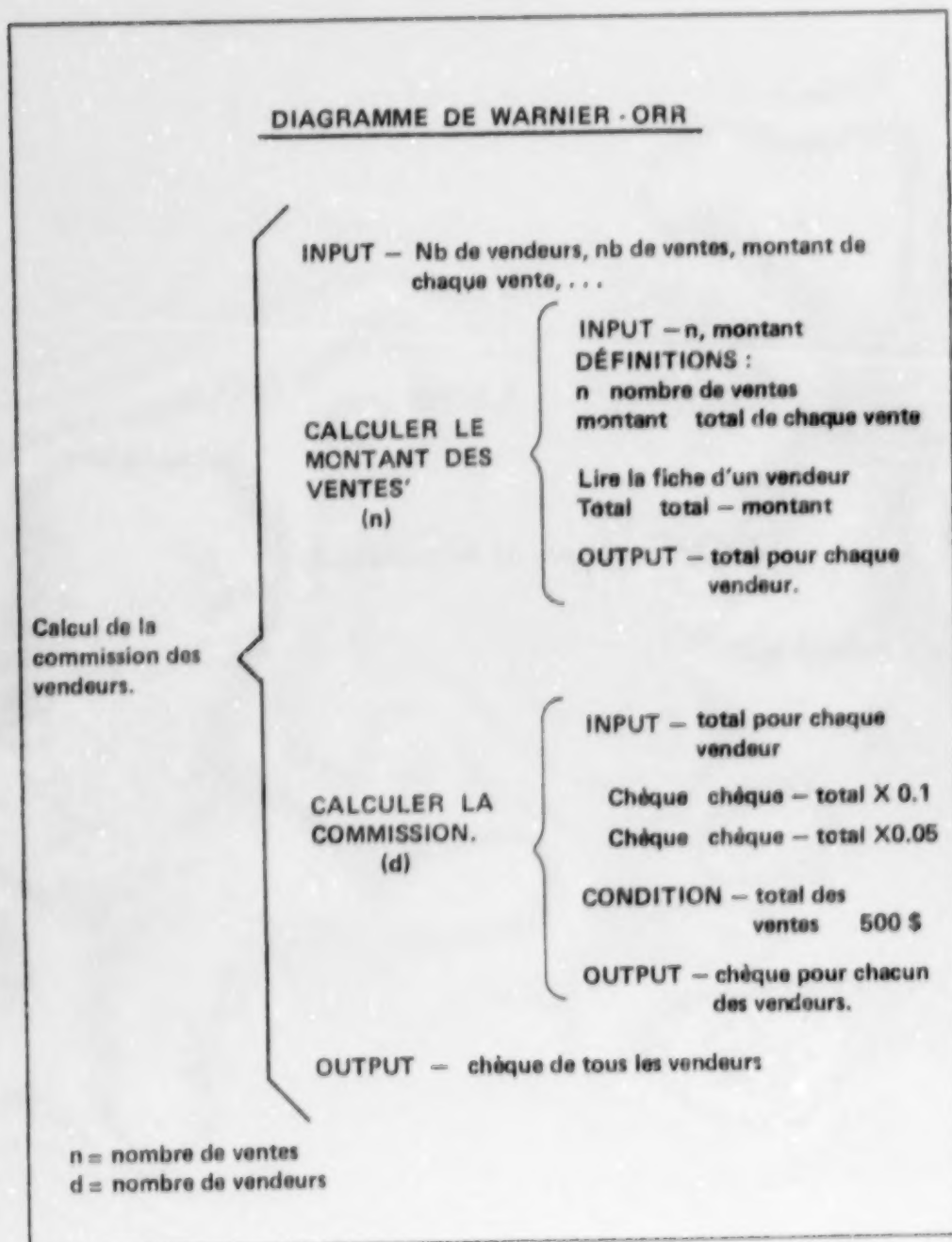


FIG. 16

MODULE 2

L'ÉLABORATION DE PROJETS

OBJECTIF TERMINAL

S'approprier des outils utiles à l'élaboration de projets

OBJECTIFS INTERMÉDIAIRES

- 2.1 Connaître les différentes étapes de l'élaboration d'un projet
- 2.2 Prendre connaissance d'un exemple de projet élaboré
- 2.3 Élaborer un projet selon ces mêmes étapes

CONTENU

- A. Textes pertinents sur les étapes d'élaboration de projets
- B. Exemple de projet élaboré
- C. Coquille pour l'élaboration d'un projet

DURÉE SUGGÉRÉE: 2 jours

INSTRUMENTATION

A. ACTIVITÉ D'APPRENTISSAGE

- Étapes d'élaboration de projets
- Exemple détaillé de projet
- Période de questions et discussion
- Choix et élaboration d'un projet
- Période d'échanges sur les projets conçus

B. MATRICES DE TRANSPARENTS

- Figures encadrées susceptibles d'être reproduites à des fins pédagogiques

C. MATÉRIEL DIDACTIQUE

- Langages de programmation
- Guides d'utilisation de ces langages
- Traitement de textes
- Logiciel de graphisme
- Ordinographe

D. DOCUMENTATION COMPLÉMENTAIRE

- Listes des projets disponibles
- Volumes spécialisés portant sur la programmation (c.f. bibliographie générale).

ÉTAPES D'ÉLABORATION DE PROJETS

1. L'énoncé du projet
2. La définition du projet
3. L'analyse
4. La spécification
5. La réalisation
6. Les tests
7. L'entretien

FIG. 1

PROGRAMMATION STRUCTURÉE

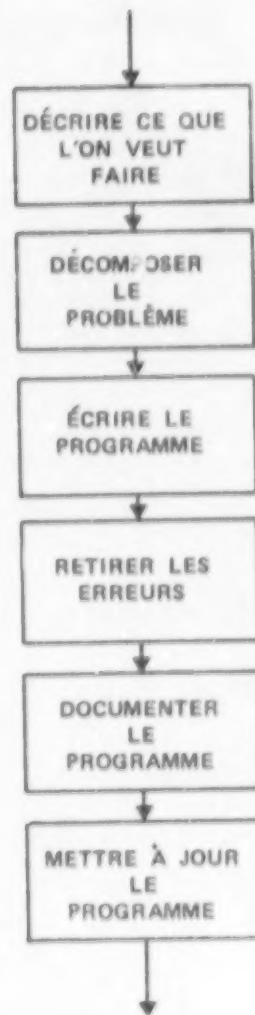


FIG. 2

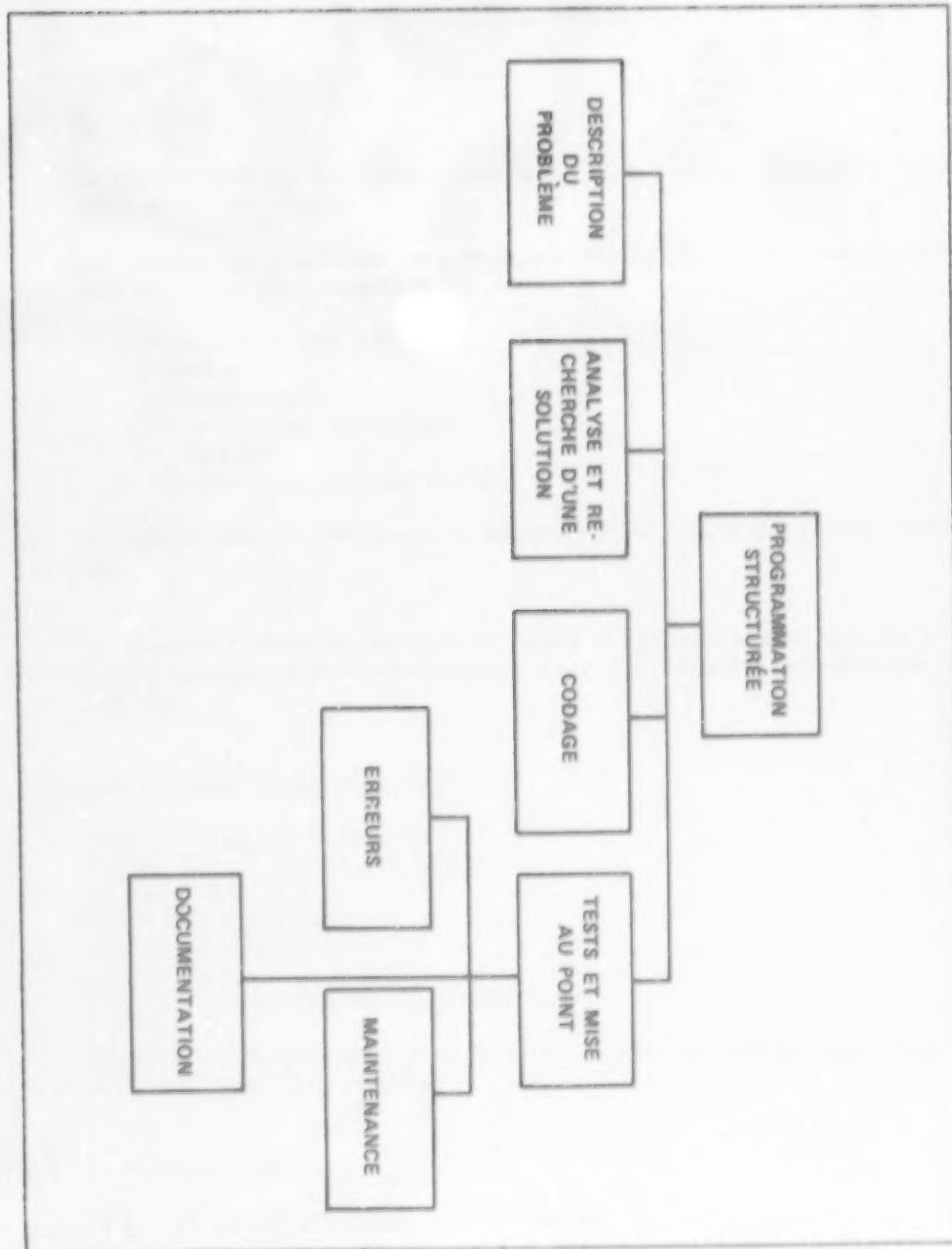


FIG. 3

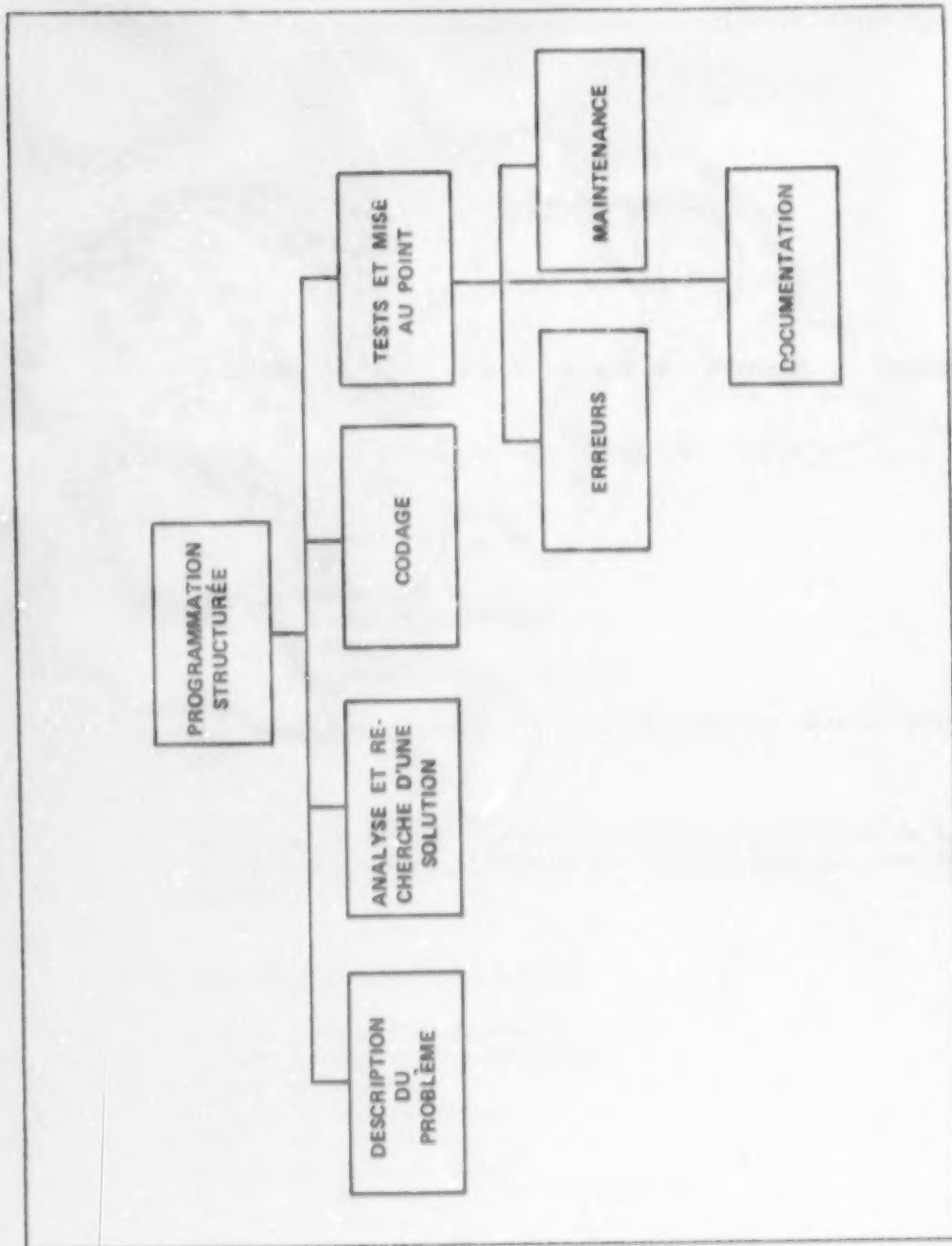


FIG. 3

L'APPROCHE-PROJET

2.1 Le projet

Le projet est un problème vaste que l'élève veut réaliser. C'est un défi qu'il se donne.

Les étudiants possèdent des talents multiples qu'ils vont mettre à profit. Ils sont capables de:

- produire des idées
- créer
- communiquer
- prévoir et anticiper
- décider
- planifier, ordonner, etc.

L'acquisition d'habiletés va se concrétiser dans un projet personnel.

Le document suivant fournit un cadre d'élaboration de projet. L'approche-projet vise à développer, chez l'étudiant, une méthode de travail.

2.2 Les grandes étapes du projet

- 2.2.1 L'énoncé du projet
- 2.2.2 La définition du projet
- 2.2.3 L'analyse
- 2.2.4 La spécification
- 2.2.5 La réalisation
- 2.2.6 La mise au point
- 2.2.7 La maintenance ou l'entretien

Voyons maintenant de façon plus détaillée les différentes étapes de l'élaboration d'un projet.

2.3 L'énoncé du projet

Énonciation de la tâche à informatiser

ex.: Je veux faire un jeu de dés!

Définition de la phase

ÉNONCÉ DU PROJET:

L'affirmation du désir de traiter de
l'information par ordinateur par une personne
(ou un groupe de personnes) qui a (ont) un
intérêt commun pour mettre en oeuvre le
processus d'informatisation.

Définition de la phase

DÉFINITION DU PROJET:

C'est l'activité qui consiste à circonscrire et à décrire de façon aussi précise que possible le travail humain ou le processus scientifique à informatiser.

2.4 La définition du projet

C'est la définition précise de ce que l'on veut faire.

- 2.4.1 S'assurer que l'on comprend bien le problème
- 2.4.2 Simuler le déroulement
- 2.4.3 Préciser les données à fournir à l'ordinateur
- 2.4.4 Préciser les résultats que l'on veut obtenir
- 2.4.5 Prévoir les différentes pages-écrans.

2.5 L'analyse du projet

L'analyse consiste à faire l'inventaire précis de toutes les actions et de toutes les décisions nécessaires pour la réalisation du projet.

Il faut:

- 2.5.1 Diviser le projet en une série de tâches
- 2.5.2 Hiérarchiser les tâches (utilisation du structurogramme ou de l'arbre algorithmique)

Un jeu peut prendre la forme suivante:

Début de la partie

- Affichage de la présentation
- Détermination du degré de difficulté

Déroulement de la partie

Affichage des résultats

N.B.: RAFFINEMENT par subdivisions

L'approche descendante aura pour effet de:

- Analyser le rôle de chaque élément du programme et du mode de calcul
- Saisir le rôle d'éléments de plus en plus petits
- Saisir l'effet de chaque instruction sur chaque variable ainsi que le but visé
- Regrouper les instructions ayant une même fonction:
MODULE
- Regrouper les modules

2.6 La spécification

Processus qui consiste à replacer dans un ordre logique toutes les étapes du projet

- 2.6.1 Diviser les tâches en sous-tâches
- 2.6.2 Morceler en une suite de niveaux d'abstraction qui vont en se précisant
- 2.6.3 Faire l'arbre algorithmique
- 2.6.4 Décrire chaque étape en pseudo-langage

Recherche d'un algorithme:

- Liste d'instructions qui nous indique quoi faire
 - . ce qu'il faut faire
 - . l'objet de l'opération
- L'algorithme ressemble à une proposition
 - . un verbe qui désigne ce qu'il faut faire
 - . un complément qui est l'objet de l'action

Critères d'un bon algorithme:

- À la fin de chaque étape de l'exécution nous pouvons identifier l'étape suivante
- Un point de départ clairement défini et un ou plusieurs points d'arrivée
- Nombre fini d'étapes
- Opérations élémentaires claires et précises (opérations les plus complexes que l'homme ou l'ordinateur peut faire sans avoir à les scinder).

2.7 La réalisation

La réalisation consiste à traduire les spécifications en langage informatique exécutable par l'ordinateur. C'est la codification dans un langage de haut niveau.

Codification:

- 2.7.1 Traduire le pseudo-code
- 2.7.2 Traduire selon les règles de syntaxe du langage utilisé

Définition de la phase

SPÉCIFICATIONS:

Processus qui consiste à replacer dans un ordre logique toutes les étapes du projet. Ordinogramme et/ou pseudo-langage général qui décrit la situation globale.

MÉTHODES DE DESCRIPTION

SPÉCIFICATION

Basée sur les données:

- WARNIER-ORR

Basée sur les structures:

- ORDINOGRAMME

- PSEUDO-LANGAGE

- ARBRE ALGORITHMIQUE

2.8 La mise au point

Il est rare qu'un programme roule du premier coup. L'étape des tests va nous permettre de mettre au point le programme en corrigeant:

- les erreurs
- les fautes
- les défauts

Mise au point:

2.8.1 Vérifier si le programme fonctionne bien dans tous les cas possibles

2.8.2 Faire un plan de test

2.8.2.1 Stratégie de mise au point

- vérifier avec des données valides
- filtrage ou mesures préventives
- vérifier les cas limites
- vérifier les formats de sortie

2.8.2.2 Le jeu d'essai

Suite à donner à la stratégie de mise au point en choisissant un ensemble de données qui vérifie le fonctionnement global du programme.

2.8.2.3 Simulation sur papier

2.8.2.4 Outils d'aide à la mise au point:
TRON, TROFF, ON ERROR GOTO ...

2.9 La maintenance ou l'entretien

Après une période de rodage, il arrive souvent que l'on veuille apporter des améliorations au programme. S'il est bien construit il sera facile de faire des modifications.

Dans cette étape il faut:

- 2.9.1 Corriger les parties qui ne fonctionnent pas
- 2.9.2 Eliminer les parties inutiles
- 2.9.3 Ajouter de nouvelles parties qui augmentent l'efficacité du programme ou ajoutent des fonctions nouvelles

2.10 Un exemple de projet

L'étudiant s'engage dans la réalisation d'un projet.

- choisir un projet
- planifier une solution
- codifier les instructions
- vérifier l'exécution
- faire des corrections et des modifications
- fournir un rapport

LES "BUGS"

ERREUR: C'est la différence entre
deux valeurs

FAUTE : Mauvaise réalisation qui
cause une erreur

DÉFAUT: C'est l'absence de ce qui
est nécessaire

RAISONS D'EFFECTUER L'ENTRETIEN

20 % 1. Corriger des fautes ou des défauts

2. Améliorer la conception

3. Améliorer l'environnement
(matériel, logiciel)

80 % 4. Relier le logiciel à
d'autres programmes

5. Effectuer des changements
dans les fichiers

6. Évolution du projet

2.10.1 Le choix du projet

- 2.10.1.1 Consulter une liste de projets.
- 2.10.1.2 Consulter une banque de projets des années antérieures sur disquettes.
- 2.10.1.3 Consulter des exemples de projets documentés (document de travail, phases d'élaboration, compte-rendu, programme sur disquette).
- 2.10.1.4 Faire le point i.e.
 - Existe-t-il quelque chose de semblable?
 - Si oui, est-il possible d'améliorer le produit, de faire différemment...?
- 2.10.1.5 Rechercher de la documentation.
 - Volumes d'informatique, de résolution de problèmes...
- 2.10.1.6 Acquérir des informations sur le sujet
- 2.10.1.7 Se familiariser avec le sujet et rechercher
 - comment animer des objets
 - comment tracer une ligne
 - comment créer l'illusion de la troisième dimension
- 2.10.1.8 Discuter avec le professeur et lui faire part de son choix
- 2.10.1.9 Discuter avec les autres étudiants.
- 2.10.1.10 Communiquer ses intentions aux autres.
- 2.10.1.11 Analyser les suggestions qui nous ont été faites.

2.10.2 L'énoncé du projet

C'est la formulation de la tâche à informatiser. J'ai une vague idée d'un projet à réaliser.

Tâche: Je veux calculer le salaire des employés.

2.10.2.1 Formuler des hypothèses de travail

Possibilités d'afficher le salaire régulier et supplémentaire.

Rechercher des exemples de projets semblables.

2.10.2.2 Formuler une ébauche de projet

Explications à fournir aux utilisateurs.
Graphisme et animation...

2.10.3 La définition du projet

C'est la définition précise des grandes étapes du projet à réaliser.

2.10.3.1 Faire un scénario

Déroulement:

- Entrer le numéro matricule de l'employé, le nombre d'heures travaillées et le salaire horaire.

Afficher:

- le numéro
- le salaire régulier
- le salaire supplémentaire
- le salaire total

2.10.3.2 Établir la liste des principales étapes.

2.10.3.2.1 Entrée des valeurs pertinentes.

2.10.3.2.2 Calculs à faire.

2.10.3.2.3 Sortie des résultats: Affichage du salaire.

2.10.4 L'analyse du projet

Le structurogramme est l'outil privilégié pour visualiser les différentes étapes du projet.

2.10.4.1 Présentation graphique

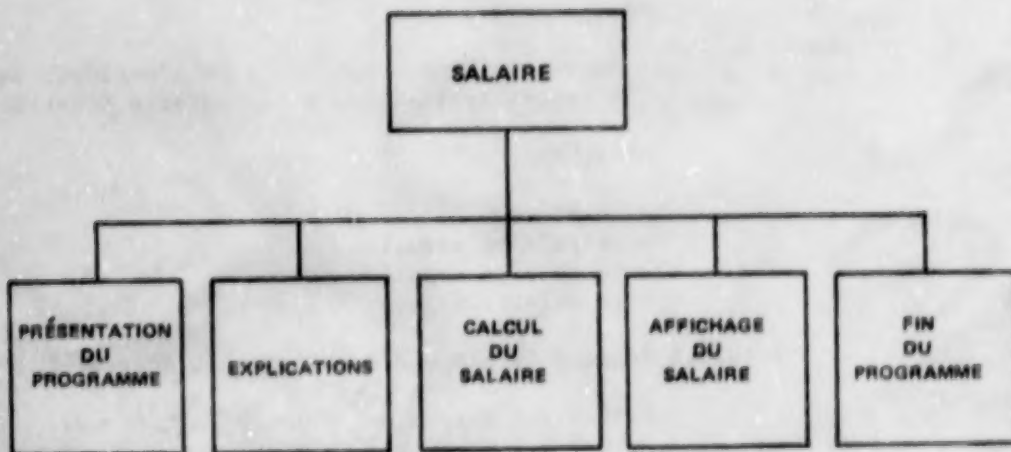
2.10.4.2 Explications

2.10.4.3 Calcul du salaire

2.10.4.4 Afficher le salaire

2.10.4.5 Fin du programme

Le salaire des employés



STRUCTUROGRAMME

2.10.5 La spécification

2.10.5.1 Décrire chacune de ces étapes en pseudo-langage et/ou sous forme d'ordinogramme.

2.10.5.2 Replacer dans un contexte logique toutes les étapes nécessaires au projet (ordino et/ou pseudo-langage général du projet)

SOUS-PROGRAMMES

CALCUL DU SALAIRE

Début

Ecrire ("Entrez le nombre d'heures travaillées et le salaire horaire séparés par une virgule . . .")

Lire (NHEURE, SHORAIRE)

SI NHEURE > 40

ALORS

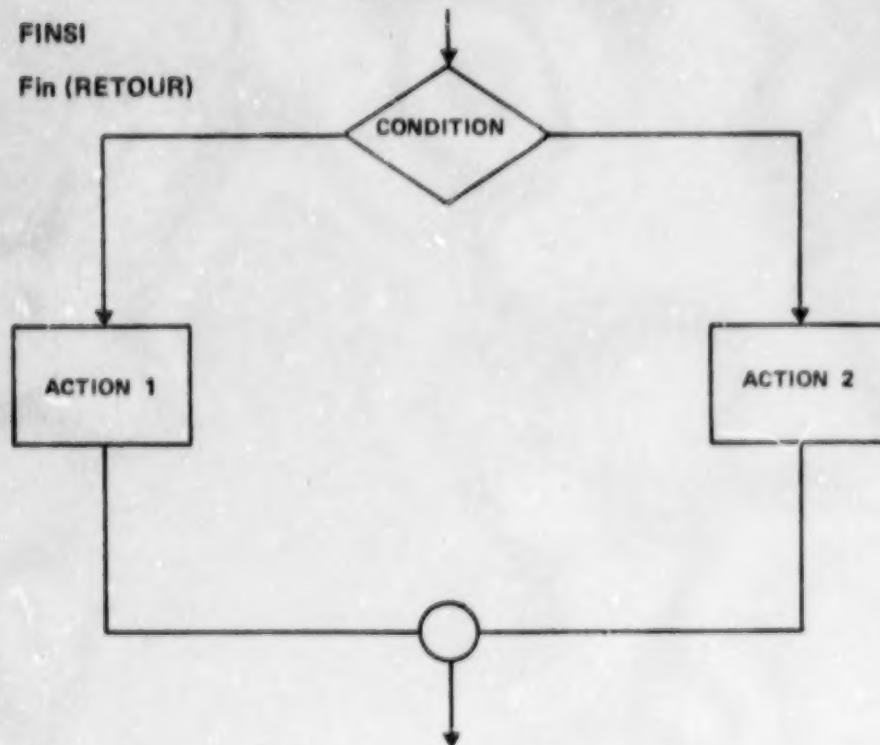
$STOTAL \leftarrow 40 \times SHORAIRE + (NHEURE - 40) \times SHORAIRE \times 1.5$

SINON

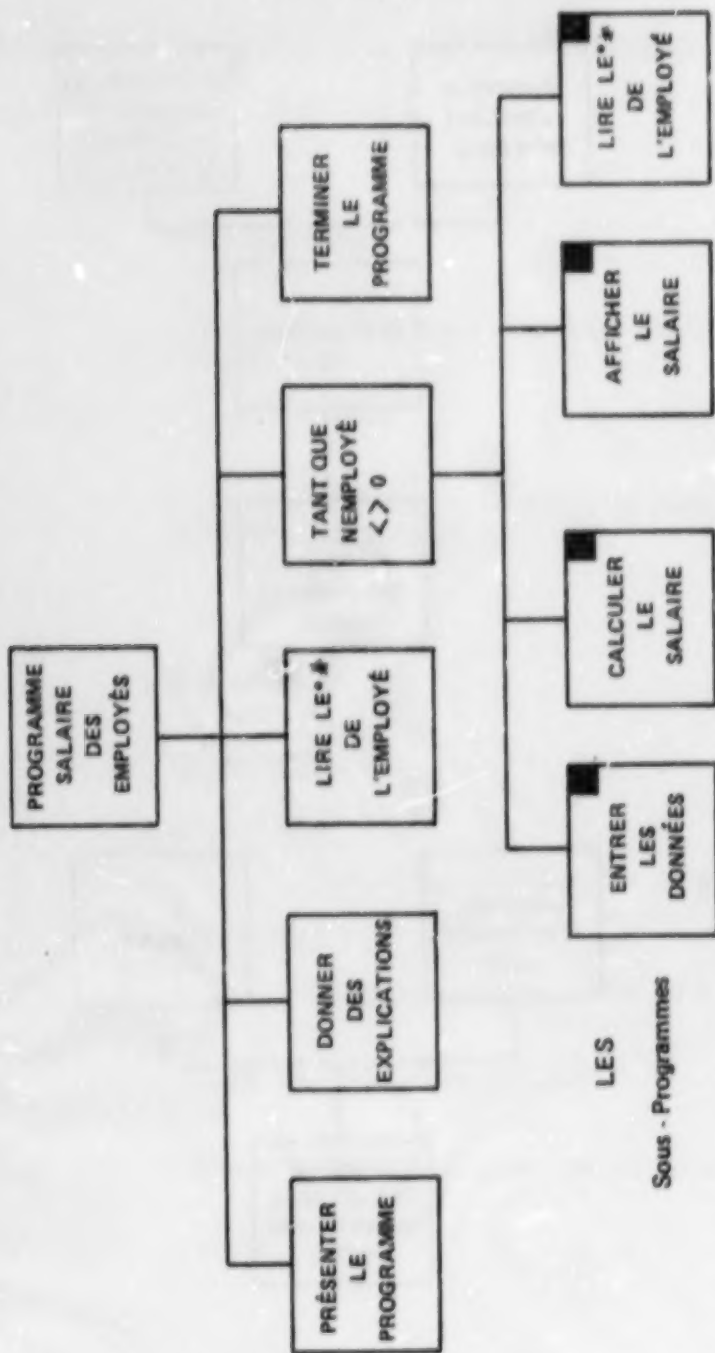
$STOTAL \leftarrow NHEURE \times SHORAIRE$

FINSI

Fin (RETOUR)

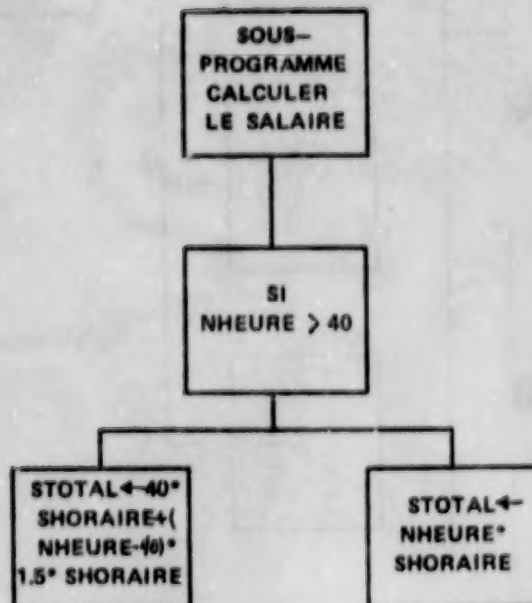
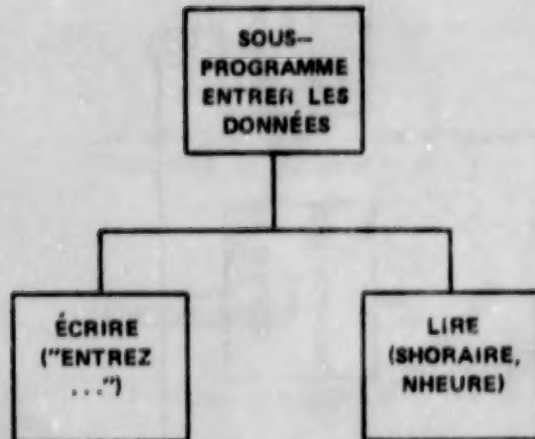


ARBRE ALGORITHMIQUE



LES
Sous-Programmes

ARBRES ALGORITHMIQUES
DES SOUS-PROGRAMMES



2.10.6 La réalisation

Codifier chacune des étapes dans un langage de programmation.

2.10.7 La mise au point

Il est important à ce stade-ci de bien vérifier la validité du programme.

Plan de tests

- Prévoir des cas

. cas 1

35 heures, taux horaire 20 \$

. cas 2

50 heures, taux horaire 20 \$

- Faire les calculs à la main

- Tester le programme avec les mêmes valeurs

2.10.8 La maintenance ou l'entretien

Si les tests ont fait ressortir des anomalies, il faudra apporter des modifications. Ces modifications seront d'autant plus facile à effectuer que notre programme est bien construit et documenté.

Les pages qui suivent contiennent des gabarits utiles: (section 2.11)

1 - fiche descriptive du projet

2 - des pages-écran

- texte : 40 colonnes
80 colonnes

- graphique : moyenne résolution
haute résolution

3 - fiche personnelle sur les étapes d'élaboration de projet

2.11 La fiche descriptive du projet

Titre du programme: _____

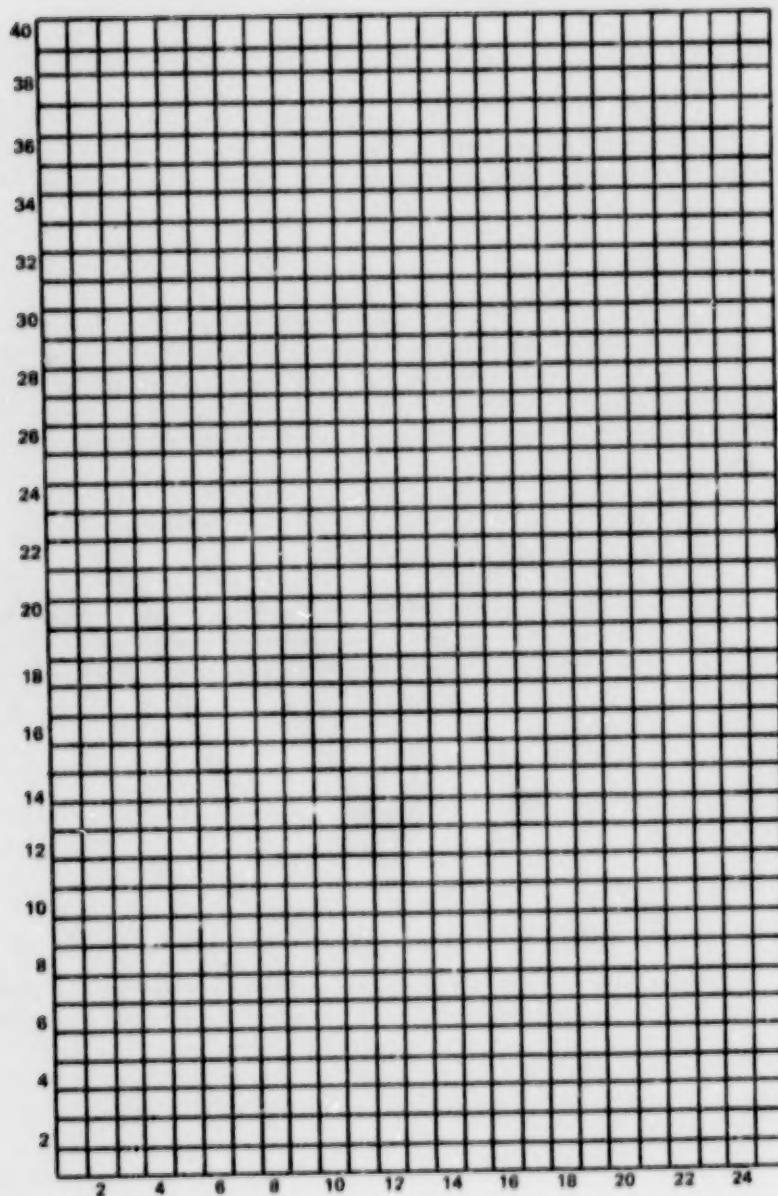
Brève description: _____

Nom des auteurs:
1. _____
2. _____
3. _____

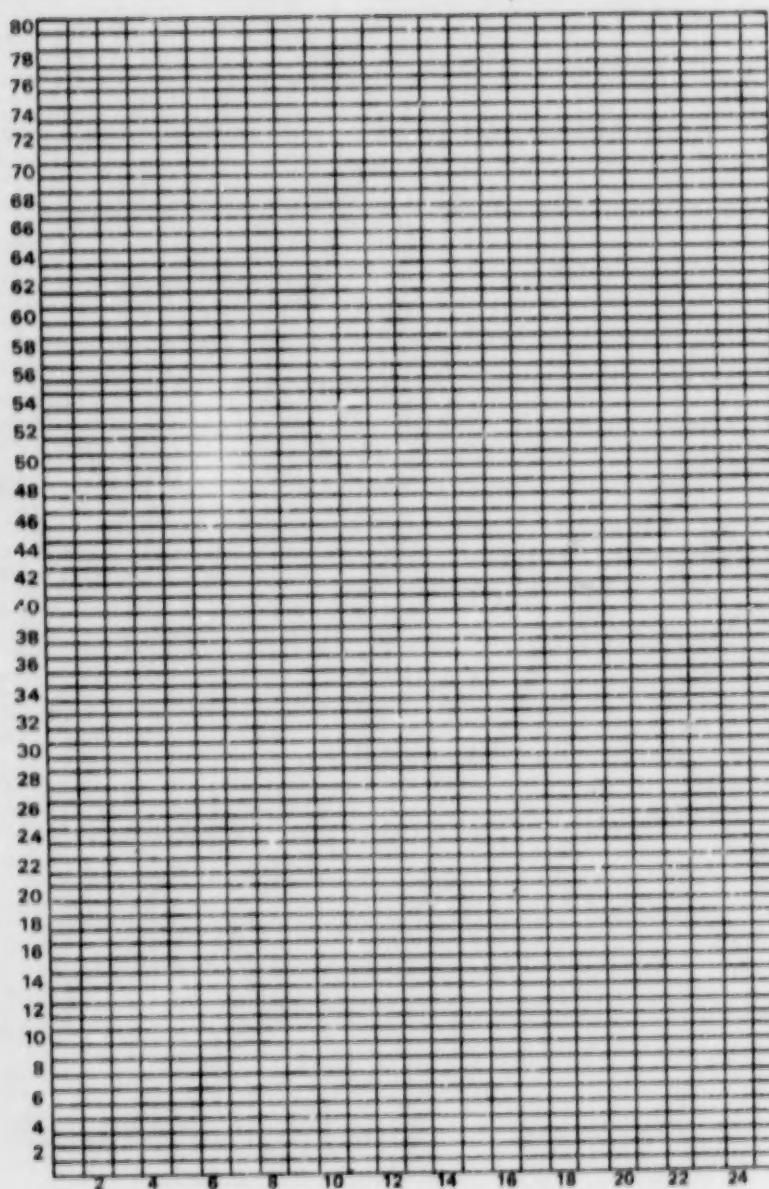
Ordinateur utilisé: _____

Périphériques nécessaires ou optionnels:

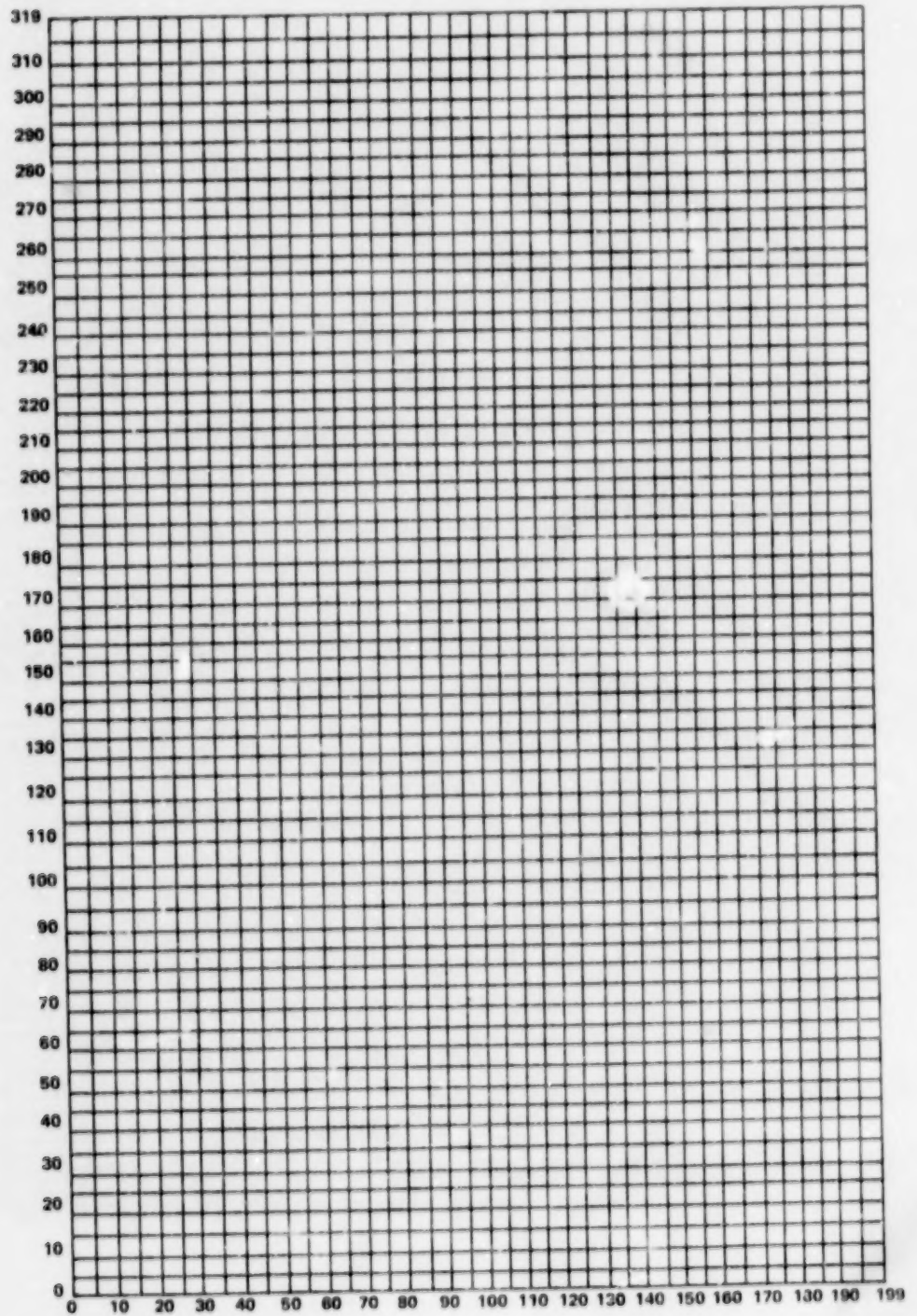
Écran-texte 40



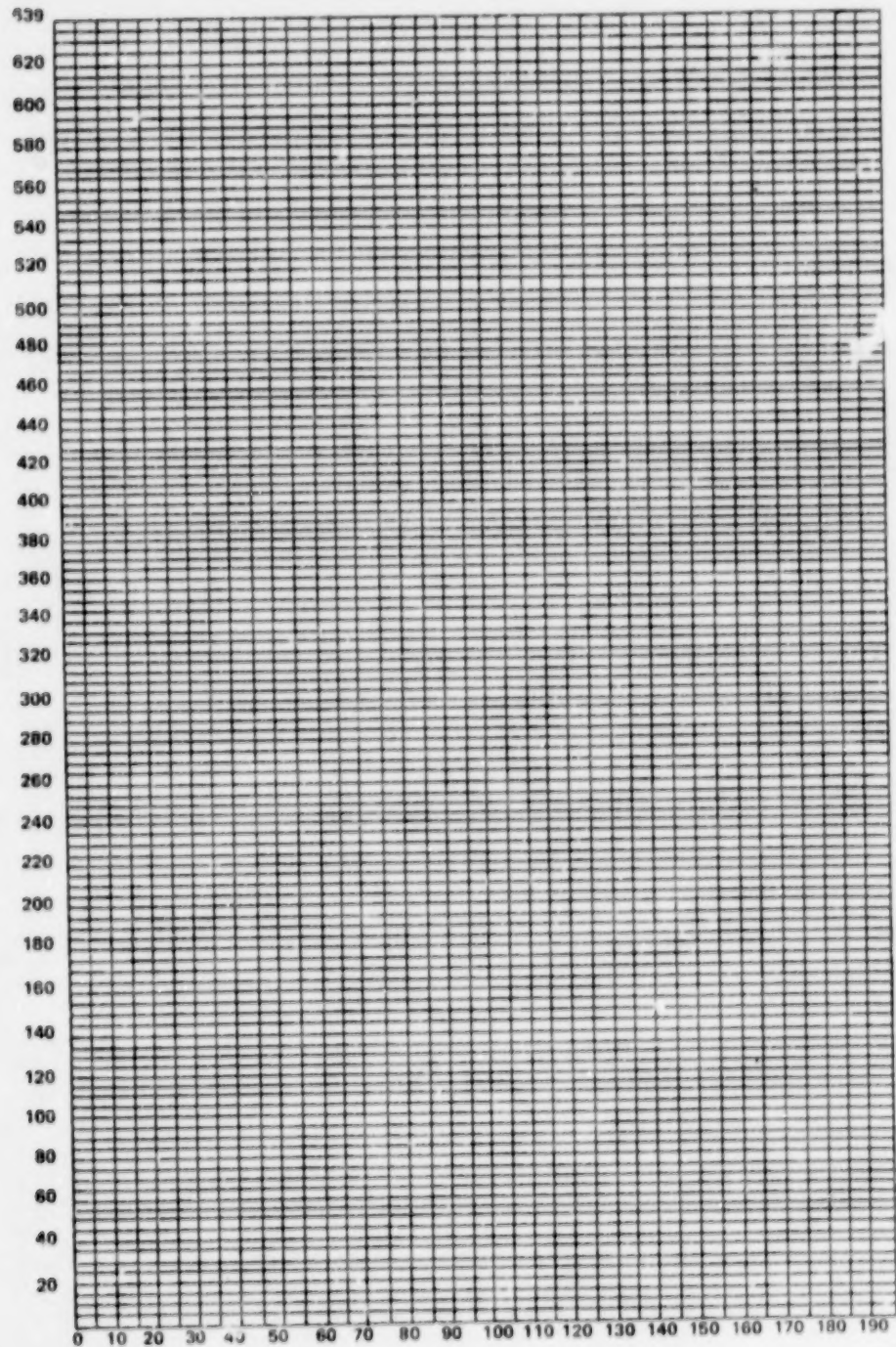
Écran-texte 80



Écran-graphique (résolution moyenne)



Écran-graphique (haute résolution)



Fiche personnelle

Élaboration du projet

1) ÉNONCÉ DU PROJET:

2) DÉFINITION DU PROJET: (faire une simulation)

N.B.: Prévoir les entrées, les traitements et les sorties.

Ces pages de présentation peuvent être préparées avec un traitement de texte. L'étudiant qui utilise ce traitement de texte complète chacune des étapes.

Fiche personnelle

Élaboration du projet

3) ANALYSE:

a) connaissance du matériel et des logiciels
disponibles

b) faire l'inventaire des actions à accomplir
faire le structurogramme et/ou l'arbre
algorithmique

Fiche personnelle

Élaboration du projet

4) LES SPÉCIFICATIONS (suite):

- b) arbre algorithmique et pseudo-langage de chaque module (sous-routine).

Fiche personnelle

Élaboration du projet

5) LA RÉALISATION: Traduire les spécifications en langage informatique, exécutable par l'ordinateur.

a) faire un tableau complet des constantes et des variables.

b) coder en langage de programmation, insérer la documentation-programme et la documentation-utilisateur.

Fiche personnelle

Élaboration du projet

- 6) Préparer les tests pour déceler les erreurs, les fautes et les défauts.

- 7) L'ENTRETIEN DU LOGICIEL
(corrections des fautes et des défauts et modifications pour rencontrer de nouvelles exigences)

MODULE 3

LA GESTION DE PROJETS

OBJECTIF TERMINAL

Gérer un projet étudiant

OBJECTIFS INTERMÉDIAIRES

- 3.1 Se familiariser avec certains concepts
- 3.2 Déterminer les étapes d'élaboration d'un projet
- 3.3 Établir les tâches se rattachant à chacune des étapes
- 3.4 Déterminer un calendrier des opérations
- 3.5 Se familiariser avec un logiciel de gestion de projets
- 3.6 Utiliser ce logiciel pour établir la répartition dans le temps

CONTENU

- A. Définitions de certains concepts
- B. Démarche d'utilisation du logiciel HPM
- C. Exemples d'utilisation de certaines représentations graphiques

DURÉE SUGGÉRÉE: 1/2 journée

INSTRUMENTATION

A. ACTIVITÉ D'APPRENTISSAGE

- Mots-clés
- Utilisation du logiciel de gestion

B. MATRICES DE TRANSPARENTS

- Figures encadrées susceptibles d'être reproduites à des fins pédagogiques

C. MATÉRIEL DIDACTIQUE

- Logiciel de gestion de projets HPM
- Guide d'utilisation de ce logiciel

D. DOCUMENTATION COMPLÉMENTAIRE

- Guide d'utilisation de Mac Project pour Macintosh
- "Software Project Management step by step", Milton D. Rosenau Jr., Marsha D. Lewin Lifetime Learning Publications
Ten Davis Drive
Belmont, California 94002

3.1 Problématique

Il faut planifier le projet que les étudiants vont entreprendre.

En effet la planification des étapes d'élaboration de projet fait partie de l'apprentissage d'une méthode de travail. Il importe à chaque enseignant de connaître un tel logiciel et de voir comment son utilisation peut servir à l'élaboration du projet. Sans vouloir généraliser un tel outil, il nous semble souhaitable que l'enseignant et l'enseignant s'en servent pour gérer les activités de l'année. A vous d'expérimenter ce logiciel et de trouver des applications intéressantes à son utilisation.

3.2 Outils de gestion de projet

3.2.1 P.E.R.T. (Project Evaluation and Review Technics 1950)

Le programme PERT permet de calculer les "optimum" d'un réseau de contraintes portant sur les tâches successives d'un chantier.

3.2.2 C.P.M. (Critical Path Method 1957)

3.2.3 Mac Project (logiciel utilisé sur Macintosh)

3.2.4 HPM (Harvard Project Management utilisé sur IBM et compatibles)

3.3 Pourquoi utiliser un tel logiciel?

Un tel logiciel permet de:

3.3.1 Visualiser les tâches d'un projet ainsi que les interrelations entre ces différentes tâches.

3.3.2 Calculer immédiatement les effets d'un changement quelconque dans la planification du projet.

3.3.3 Décider de la meilleure façon d'utiliser les ressources.

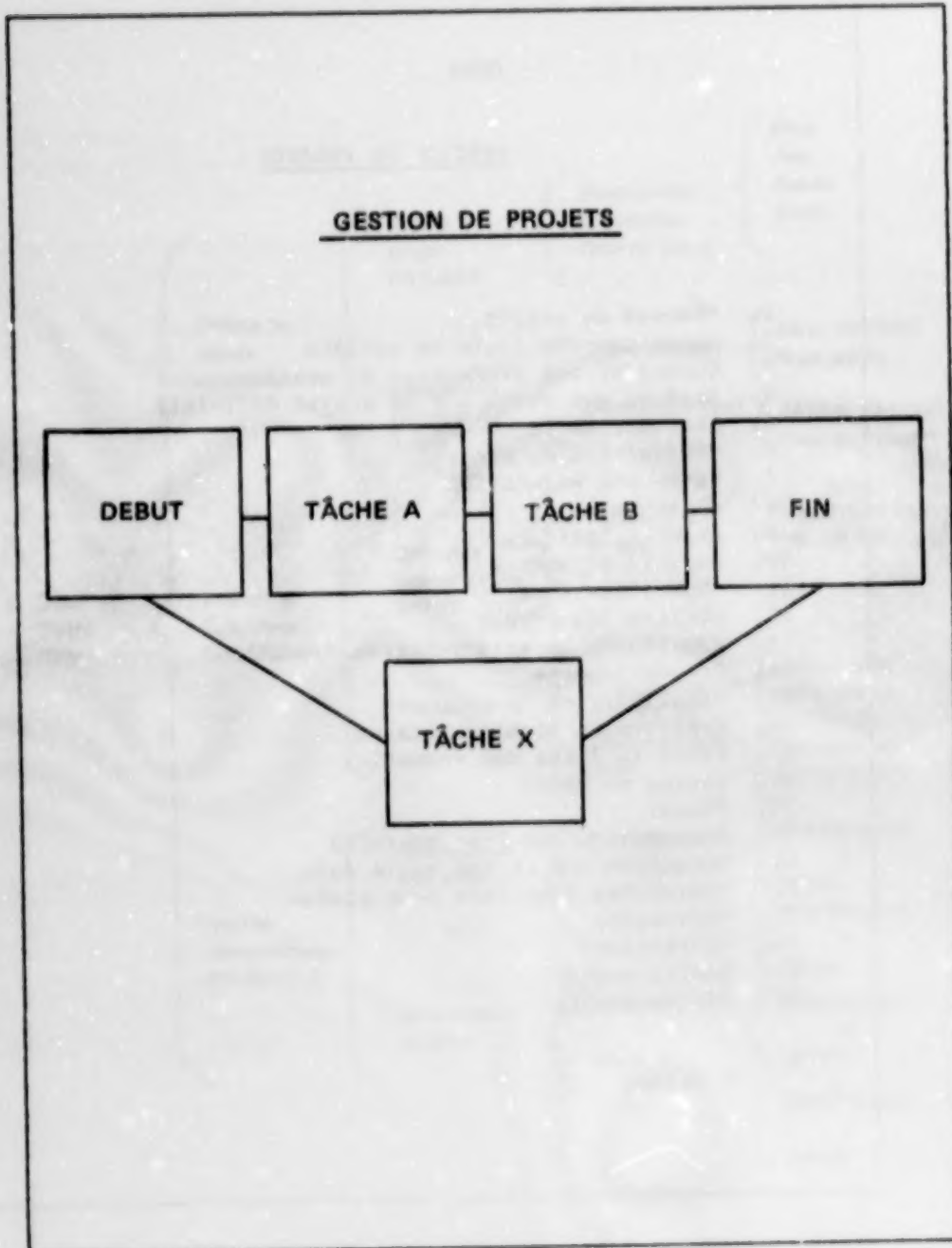
3.3.4 Communiquer aux différents intervenants les choses importantes à faire pour la réalisation du projet à l'intérieur d'une période donnée.

3.4 Concepts de base

- 3.4.1 **Projet:** ensemble d'activités interreliées directement, orientées vers un but spécifique, devant être réalisées dans un cadre temporel spécifique.
- 3.4.2 **Tâche (Task):** activité spécifique comprenant une définition précise.
- 3.4.3 **Jalon (Milestone):** événement spécifique dans le temps, qui peut être décrit objectivement.
- 3.4.4 **Chemin critique:** Trajet qui, en unissant l'ensemble des tâches représentées par un graphe, correspond au plus court temps total pour l'exécution de cet ensemble de tâches. C'est le chemin dont le flottement est nul.
- 3.4.5 **Flottement ou délai:** Différence entre la durée d'un chemin quelconque sur un graphe de tâches, et le chemin critique qui représente le plus court temps possible pour l'exécution du projet qui réunit l'ensemble des tâches en cause.
- 3.4.6 **Graphe:** C'est un instrument de la gestion logique des ressources. L'allure en réseau du graphe est donnée par le fait que plusieurs séries de tâches peuvent se développer simultanément.

3.5 Utilisation du Harvard Project Management

- 3.5.1 Charger le logiciel
- 3.5.2 Définir le projet
- 3.5.3 Schématiser les jalons et les tâches
- 3.5.4 Mémoriser le projet sur la disquette
- 3.5.5 Élaborer le calendrier de travail
- 3.5.6 Mémoriser le calendrier sur la disquette
- 3.5.7 Planifier l'horaire

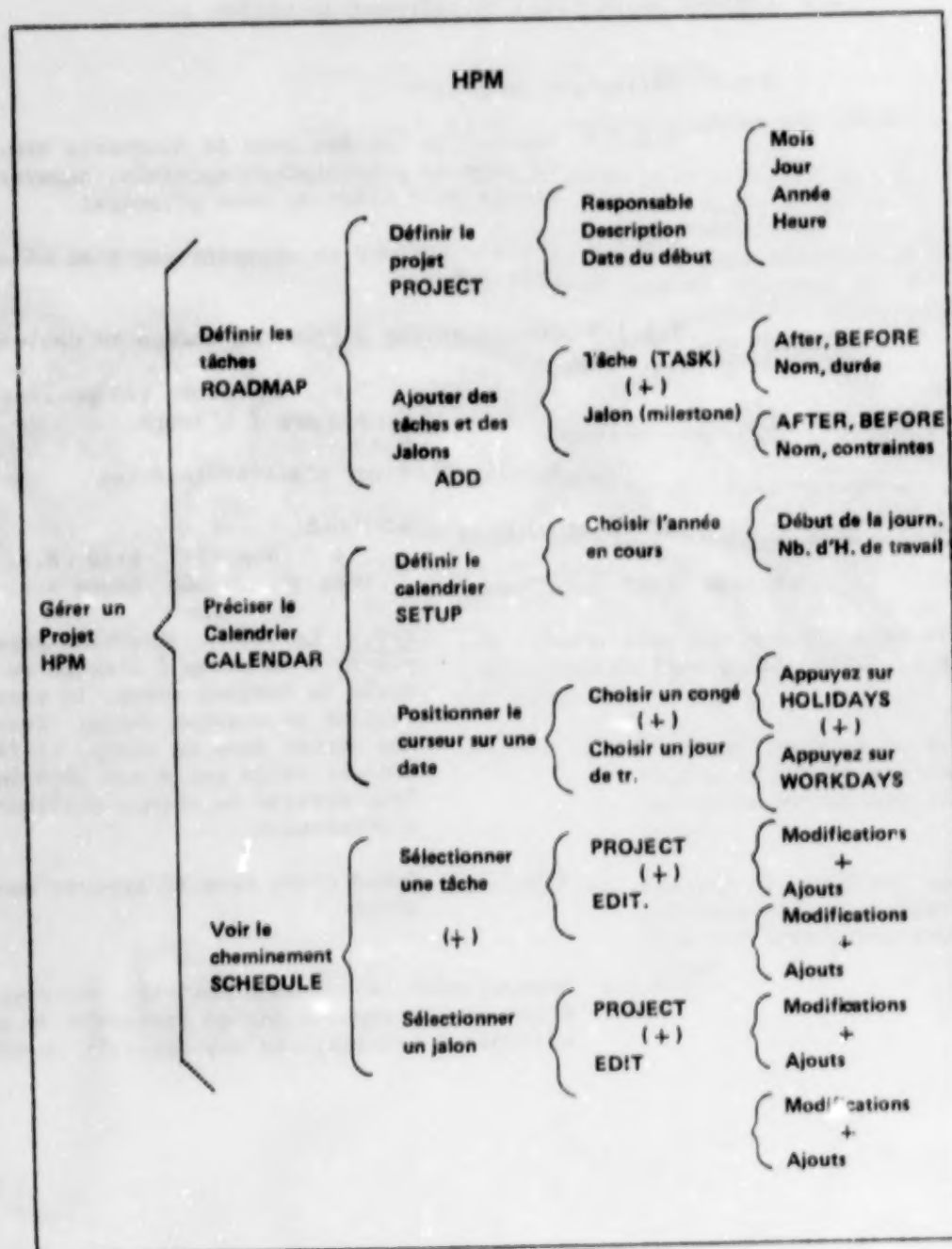


GESTION FIG. 1

GESTION DE PROJETS

1. *Énoncé du projet
2. Consulter une liste de projets
3. Consulter son professeur et ses camarades
4. Arrêter son choix sur un projet définitif
5. Chercher de la documentation
6. *Définition du projet
7. Faire une simulation
8. *Analyse
9. Aspects logiques
10. Aspects physiques
11. *Spécifications
12. Définir les étapes
13. Construire le structurogramme
14. Pseudo-langage
15. *Réalisation
16. Création du fichier interne
17. Faire la liste des variables
18. Codage en BASIC
19. *Tests
20. Commentaires aux bons endroits
21. Questions une et une seule fois
22. Toutes les questions sont posées
23. *Entretien
24. Corrections
25. Modifications
26. Fin du projet

* Jalons



GESTION FIG. 3

3.6 Charger le logiciel et préparer le projet

3.6.1 Définition du projet

3.6.1.1 Amorcer le système avec la disquette HPM. Lorsque la page de présentation apparaît, appuyer sur une touche pour aller au menu principal.

3.6.1.2 Choisir ROADMAP en appuyant sur R et sélectionner P pour PROJECT.

3.6.1.3 Compléter les différents champs du tableau

3.6.1.3.1 Utiliser la touche TAB pour se déplacer d'un champ à l'autre.

3.6.1.3.2 Façon d'entrer la date:

SCHEDULED START:

4 / 1 / 86 8:30 A.M.
mois/jour/année heure A.M.

N.B.: La clé de direction permet de passer d'un champ à l'autre de la date. Après le dernier champ, le curseur revient au premier champ. Pour inscrire une donnée dans un champ, il faut d'abord effacer celle qui y est déjà écrite. Pour effacer un champ, utilisez la clé d'effacement.

3.6.1.3.3 Quand c'est terminé, appuyer deux fois sur ENTER.

3.6.1.4 Sélectionner la commande ROADMAP. Deux jalons, START et END séparés par un rectangle de tâche, sont affichés à l'écran. Au bas apparaît le MENU.

3.6.1.5 Schématisation des tâches.

3.6.1.5.1 Choisir ADD au menu

3.6.1.5.2 Choisir TASK et appuyer sur ENTER

3.6.1.5.3 Choisir AFTER pour indiquer que la tâche nouvelle se place après le jalon START. À la demande ADD TASK AFTER START BEFORE _____ écrire END pour indiquer le jalon END et confirmer en appuyant sur ENTER.

3.6.1.5.4 Compléter les parties suivantes du tableau NODE.

TASK: _____

DESCRIPTION: _____

PLANNED/DURATION: _____

Appuyer sur ENTER deux fois.

3.6.1.5.5 Les informations sur une tâche particulière peuvent être modifiées de la façon suivante:

3.6.1.5.5.1 Utiliser les clés de direction pour se positionner sur une tâche et choisir EDIT au menu.

3.6.1.5.5.2 Ajouter ou modifier les informations et appuyer deux fois sur ENTER pour confirmer .

3.6.1.6 Schématisation des jalons.

La schématisation des jalons se fait sensiblement de la même façon.

- 3.6.1.6.1 Placer le curseur sur la dernière tâche
- 3.6.1.6.2 Choisir les commandes ADD MILESTONE
AFTER
- 3.6.1.6.3 Appuyer sur ENTER
- 3.6.1.6.4 Compléter le tableau qui apparaît

3.6.1.7 Mémorisation des informations sur la disquette

- 3.6.1.7.1 Faire FILE S PROJECT NOM
- 3.6.1.7.2 Faire QUIT pour revenir au menu principal

3.6.2 Élaboration d'un calendrier

Il s'agit maintenant d'élaborer un calendrier pour le projet nouvellement élaboré i.e. temps accordé, mois choisis, jours ouvrables, congés prévus, début de la journée et nombre d'heures travaillées.

3.6.2.1 Définition du calendrier

- 3.6.2.1.1 Lorsque qu'au menu principal, appuyer sur
CALENDAR
- 3.6.2.1.2 Utiliser la commande SETUP

3.6.2.2 Définition de l'année et du nombre d'années

- 3.6.2.2.1 Effacer le premier chiffre de l'année et inscrire 6
- 3.6.2.2.2 Se déplacer au champ suivant, effacer le 0 et inscrire 1

3.6.2.3 Définition de l'heure du début de la journée de travail et du nombre d'heures travaillées

3.6.2.3.1 Déplacer le curseur à l'aide de la clé TAB, effacer le 9 et inscrire 8

3.6.2.3.2 Passer au champ suivant et inscrire 8:30

3.6.2.3.3 Confirmer tous les choix en appuyant sur ENTER

3.6.2.4 Calendrier des jours de congés

3.6.2.4.1 Appuyez sur ENTER pour avoir les six premiers mois de l'année

3.6.2.4.2 Déplacer le curseur sur le jour désiré

3.6.2.4.3 Choisir HOLIDAY au menu

N.B.: Il est possible ainsi de prévoir au calendrier les congés statutaires et les interétapes de l'année scolaire en cours.

3.6.2.5 Sauvegarde du calendrier

Faire FILE SAVE CALENDAR

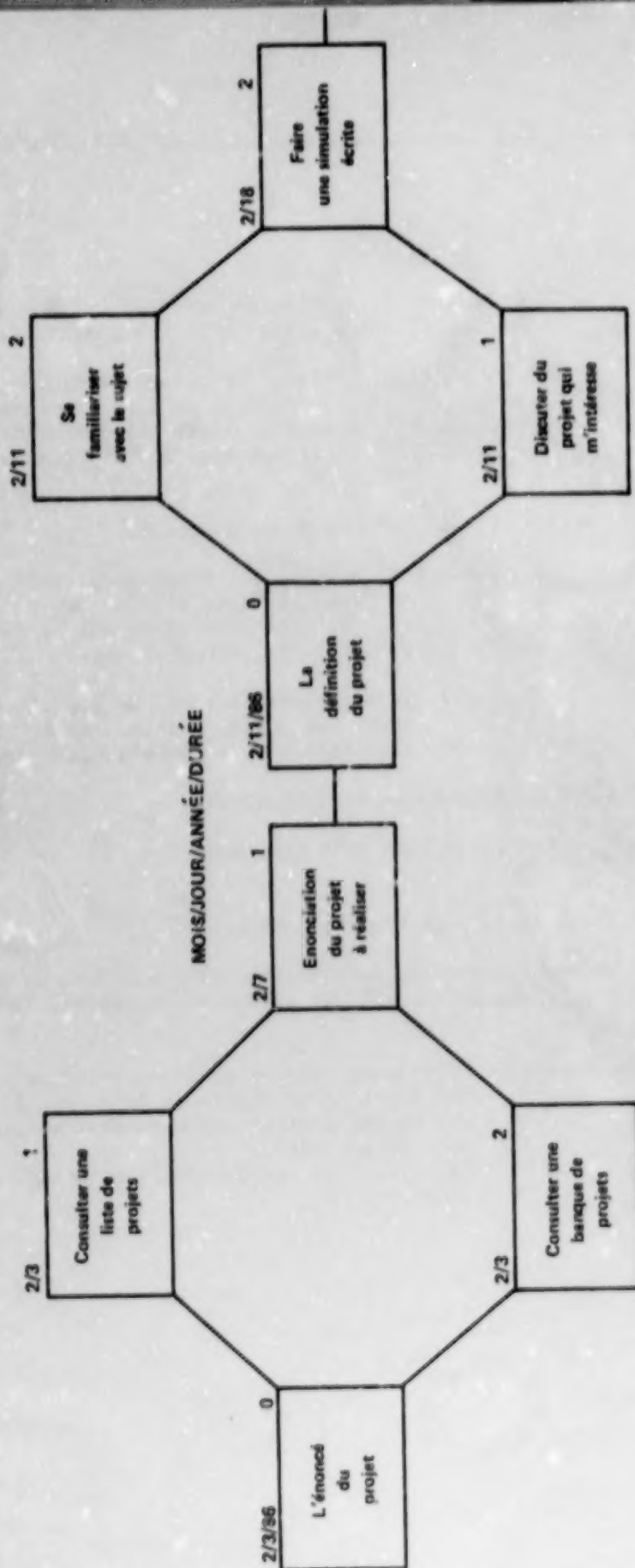
3.6.3 Planification de l'horaire du projet

Lorsque toutes les informations sont entrées, on peut obtenir un diagramme de Gantt qui permet de visualiser la planification.

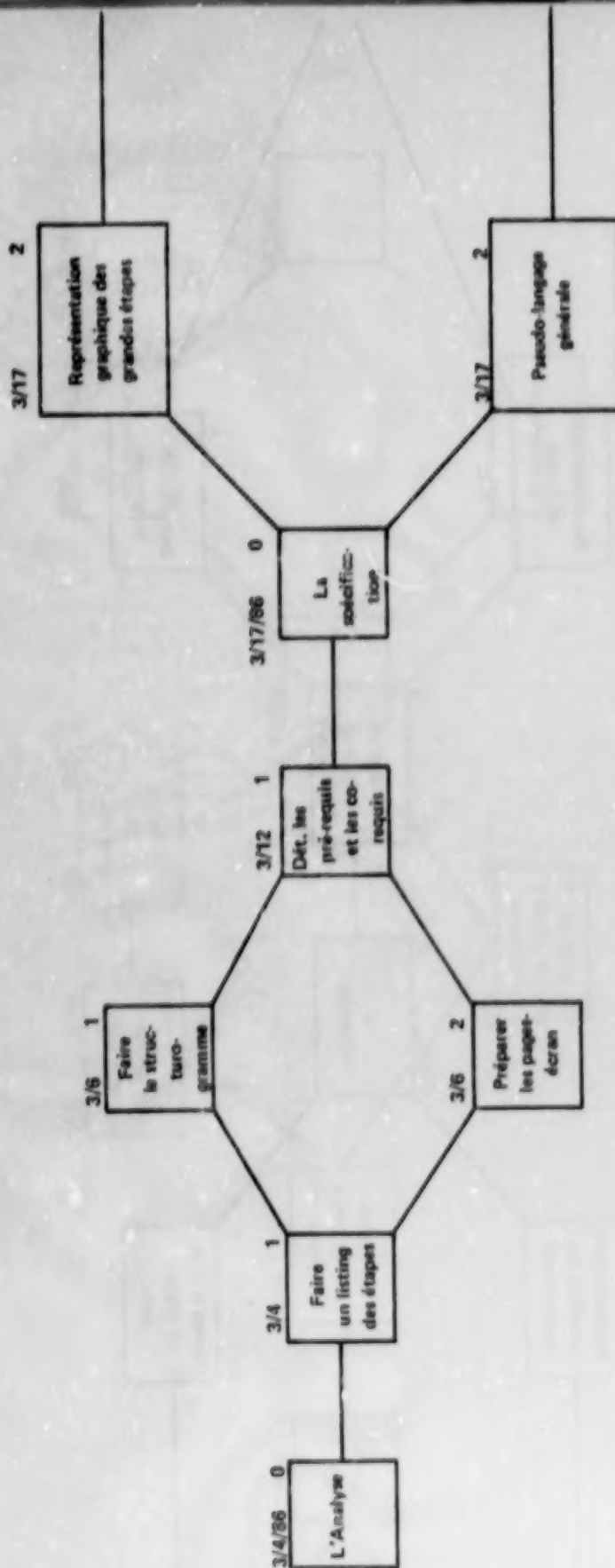
Légende:

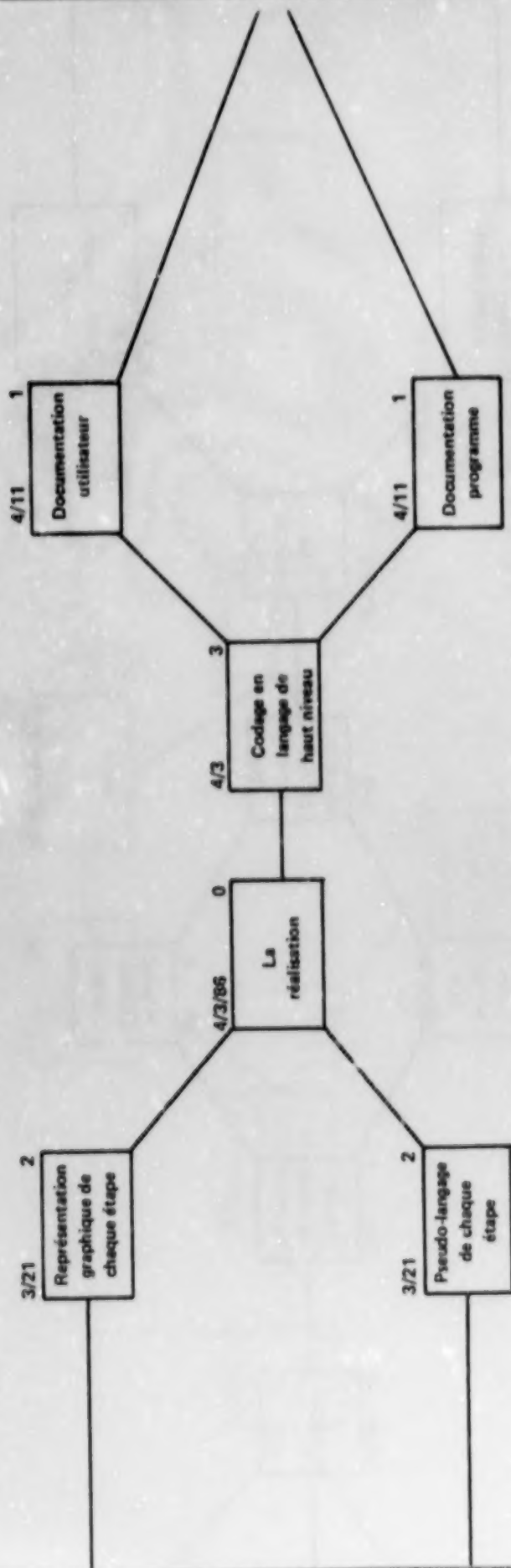
- un diamant indique un jalon critique
- une paire de triangles un jalon non critique
- les barres pleines correspondent au temps accordé pour chaque tâche
- la ligne fine indique le délai possible pour une tâche donnée

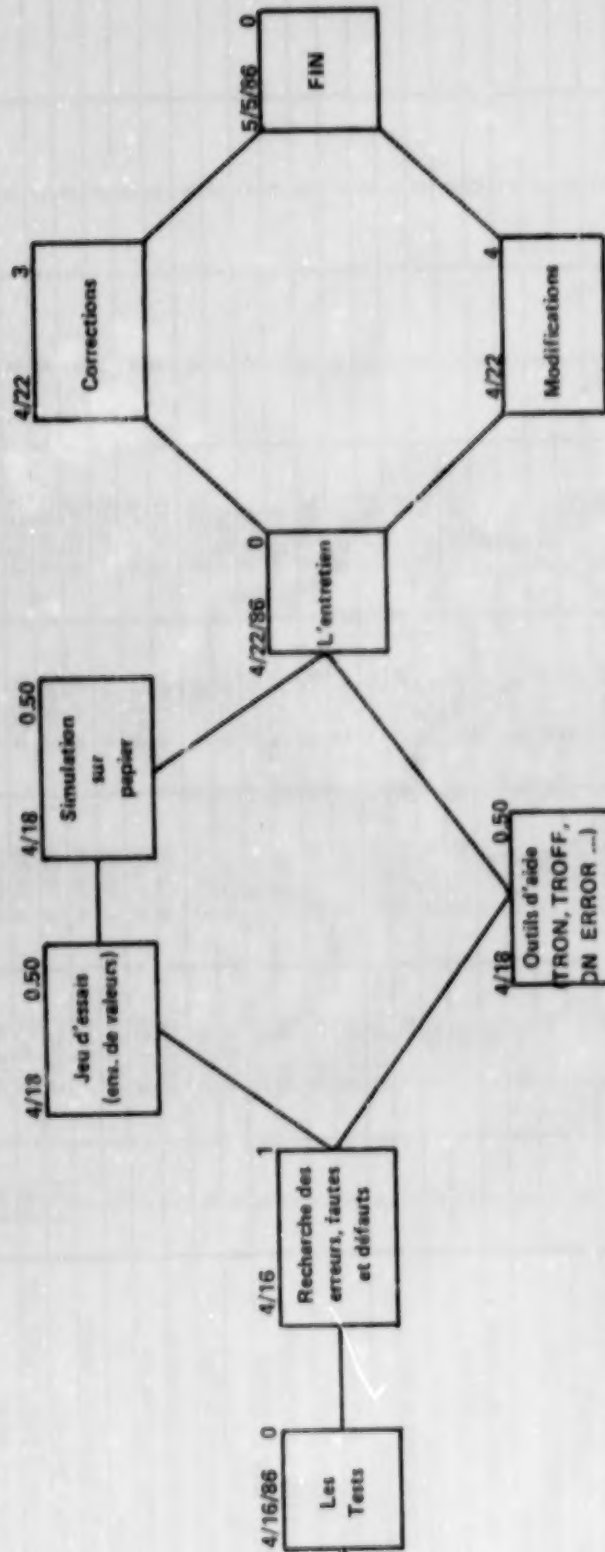
LE SALAIRE DES EMPLOYÉS (gestion de projet)



N.B.: 4 périodes sur 3 jours
1 période 1 heure







MODULE 3

SALAIRE DES EMPLOYÉS

TASK NAME	DAYS	EARLIES START	EARLIES FINISH	LATES START	LATES FINISH	FIXED COST	RESOURCE COST	FIXED INC.
1 L'énoncé du projet	0	2/3/86	2/3/86	2/3/86	2/3/86	0	0	
2 Consulter une liste de projets	1	2/3/86	2/5/86	2/5/86	2/7/86	0	0	
3 Consulter une banque de	2	2/3/86	2/7/86	2/3/86	2/7/86	0	0	
4 Enonciation du projet à	1	2/7/86	2/11/86	2/7/86	2/11/86	0	0	
5 La définition du projet	0	2/11/86	2/11/86	2/11/86	2/11/86	0	0	
6 Discuter du projet qui m'intéresse	1	2/11/86	2/14/86	2/14/86	2/18/86	0	6	
7 Se familiariser avec le	2	2/11/86	2/18/86	2/11/86	2/18/86	0	0	
8 Faire une simulation écrite	2	2/18/86	3/4/86	2/18/86	3/4/86	0	0	
9 L'analyse	0	3/4/86	3/4/86	3/4/86	3/4/86	0	0	
10 Faire un "listing" des étapes	1	3/4/86	3/6/86	3/4/86	3/6/86	0	0	
11 Faire le structuro-gramme	1	3/6/86	3/10/86	3/10/86	3/12/86	0	0	
12 Préparer les pages-écran	2	3/6/86	3/12/86	3/6/86	3/12/86	0	0	
13 Det. les pré-requis et les	1	3/12/86	3/17/86	3/12/86	3/17/86	0	0	
14 La spécification	0	3/17/86	3/17/86	3/17/86	3/17/86	0	0	
15 Représentation graphique	2	3/17/86	3/21/86	3/17/86	3/21/86	0	0	
16 Pseudo-langage générale	2	3/17/86	3/21/86	3/17/86	3/21/86	0	0	
17 Représentation graphique de	2	3/21/86	4/3/86	3/21/86	4/3/86	0	0	
18 Pseudo-langage de chaque	2	3/21/86	4/3/86	3/21/86	4/3/86	0	0	
19 La réalisation	0	4/3/86	4/3/86	4/3/86	4/3/86	0	0	
20 Codage en langage de haut	3	4/3/86	4/11/86	4/3/86	4/11/86	0	0	
21 Documentation programme	1	4/11/86	4/16/86	4/11/86	4/16/86	0	0	
22 Documentation utilisateur	1	4/11/86	4/16/86	4/11/86	4/16/86	0	0	
23 Les tests	0	4/16/86	4/16/86	4/16/86	4/16/86	0	0	
24 Recherche des erreurs, fautes	1	4/16/86	4/18/86	4/16/86	4/18/86	0	0	
25 Jeu d'essais lens. de...	0.50	4/18/86	4/18/86	4/18/86	4/18/86	0	0	
26 Outils d'aide (TRON, TROFF, ON	0.50	4/18/86	4/18/86	4/18/86	4/22/86	0	0	
27 Simulation sur papier	0.50	4/18/86	4/22/86	4/18/86	4/22/86	0	0	
28 L'entretien	0	4/22/86	4/22/86	4/22/86	4/22/86	0	0	
29 Corrections	3	4/22/86	5/1/86	4/22/86	5/5/86	0	0	
30 Modifications	4	4/22/86	5/5/86	4/22/86	5/5/86	0	0	
31 FIN	0	5/5/86	5/5/86	5/5/86	5/5/86	0	0	

MODULE 4

LA PLANIFICATION DE L'ANNÉE

OBJECTIF TERMINAL

Planifier le contenu et les interventions de l'année

OBJECTIFS INTERMÉDIAIRES

- 4.1 Planifier les premiers cours
- 4.2 Choisir un sous-ensemble de commandes-système et d'instruction-langage
- 4.3 Planifier les problèmes à résoudre
- 4.4 Planifier le fonctionnement des équipes de travail à l'intérieur du groupe

CONTENU

Textes sur:

- A. Les premiers cours
- B. Les premiers problèmes
- C. La planification des équipes de travail

DURÉE SUGGÉRÉE: 1/2 journée

INSTRUMENTATION

A. ACTIVITÉ D'APPRENTISSAGE

- Planification des cours et de leur contenu
- Repartition de la matière
- Choix des activités

B. MATÉRIEL DIDACTIQUE

- Objectifs généraux, terminaux et intermédiaires du programme d'études "Introduction à la Science de l'Informatique"
- Guide d'utilisation du langage de programmation utilisé.

C. DOCUMENTATION COMPLÉMENTAIRE

- Volume d'initiation à la programmation et comportant des exemples de programmation.

c.f. bibliographie générale à la fin de ce document.

LES PREMIERS COURS

Introduction

Comme le souligne le guide pédagogique, le goût de la machine étant beaucoup trop fort, il serait regrettable de procéder avec des cours de méthodologie de la programmation.

Cependant avant d'en arriver au comportement souhaité i.e. résolution de problèmes, méthode de travail etc., les premiers contacts avec les étudiants sont très importants et peuvent influencer beaucoup leur comportement et leur motivation à ce cours.

C'est pour cela qu'il est très important, dans un premier temps, que les étudiants s'approprient la machine le plus rapidement possible et connaissent la façon dont le professeur va fonctionner à l'intérieur du cours.

Introduction de certains concepts

Il s'agit de mettre en contact les étudiants avec la machine le plus rapidement possible. C'est un premier pas vers l'autonomie.

4.1 DESCRIPTION DE L'APPAREIL

4.1.1 La console

1.1.1.1 Le clavier

1.1.1.2 L'unité de disquettes

4.1.2 Le moniteur

4.2 LA MISE EN SERVICE DE L'APPAREIL

- 4.2.1 Insérer la disquette-étudiant
- 4.2.2 Verrouiller l'unité de disquettes
- 4.2.3 Mettre l'interrupteur à MARCHÉ
- 4.2.4 Introduire la date et valider
- 4.2.5 Introduire l'heure et valider
- 4.2.6 Attendre l'affichage désiré avant de commencer à travailler

4.3 UTILISATION DU CLAVIER

- 4.3.1 Connaissance minimale du clavier (utilisation de certaines touches spécifiques)
 - 4.3.1.1 Connaître le mode d'édition
 - 4.3.1.2 Connaître le fonctionnement des touches programmables
- 4.3.2 Matériel utilisé
 - 4.3.2.1 Exercices d'apprentissage
 - 4.3.2.2 Tutoriels pour l'utilisation du clavier du MAX

4.4 AFFICHAGE DE TEXTES ET DE VALEURS NUMÉRIQUES

- 4.4.1 Affichage de textes
 - 4.4.1.1 En mode direct
 - 4.4.1.2 En mode programmation

4.4.2 Affichage numériques

- 4.4.1 En mode direct (mode calculatrice)
- 4.4.2 En mode programmation

4.5 LES PREMIERS PROBLÈMES À RÉSOUDRE

Il s'agit ici de déterminer des exercices qui vont permettre de s'approprier la machine le plus rapidement possible. Nous choisissons donc un sous-ensemble de commandes de langage qui permettent l'affichage sur l'écran et le dialogue avec l'ordinateur.

4.5.1 Problème: Afficher un texte sur l'écran.

4.5.2 Problème: Afficher des noms et prénoms sur 2 colonnes.

4.5.3 Problème: Afficher un texte sur la 12e ligne à partir de la colonne 20.

4.5.4 Problème: Fiche individuelle avec, comme renseignements, le nom, le prénom, le foyer et l'âge.

- lire les renseignements
 - o variables numériques et alphanumériques
 - o affectations
- afficher les renseignements
 - o mise en page

4.5.5 Problème: Afficher les renseignements suivants (nom, prénom, foyer, adresse) à l'intérieur d'un encadrement quelconque.

4.6 PRÉPARATION D'UNE DISQUETTE-ÉTUDIANT

4.6.1 Utilité d'une telle disquette

Cette disquette permet à l'étudiant de travailler en mode programmation, sans avoir à passer les commandes-systèmes.

4.6.2 Mode d'utilisation

Voir 4.2, "Mise en service de l'appareil"

4.6.3 Préparation de la disquette-étudiant

Utiliser un fichier "autoexec."

4.7 FONCTIONNEMENT DU GROUPE DE TRAVAIL

4.7.1 Formation des équipes étudiantes

Permettre aux étudiants de former des équipes de travail. L'expérience tend à démontrer que le travail à deux est beaucoup plus fructueux. D'autres étudiants voudront travailler seuls.

4.7.2 Laboratoire de huit(8) appareils

Huit équipes travaillent sur les appareils tandis que les autres préparent leur travail ou écoutent des explications que vous avez à leur donner. Au milieu de la période ou selon le besoin, on inverse les rôles. Ceci permet au professeur de travailler avec des groupes plus restreints.

4.7.3 Laboratoire de seize(16) appareils

En principe les 32 étudiants peuvent aller travailler sur les appareils. Dans ce cas, le professeur, selon les besoins et les questions des étudiants, prend quelques minutes au début de la période pour expliquer le travail à faire, pour distribuer des notes polycopiées ou autres documents. Il a le loisir par la suite d'agir comme consultant, aide, dépanneur auprès des différentes équipes.

4.8 Planification

L'année scolaire ne comporte que 180 jours de classe et le cours d'I.S.I. occupe 4 périodes de 75 minutes sur un horaire de 9 jours. Il est tout à fait réaliste de prévoir approximativement 75 périodes au total pendant l'année. Nous devons planifier des activités qui permettront aux étudiantes et étudiants de réaliser pleinement les objectifs du programme d'études en débordant le moins possible du cadre temporel prévu. Des travaux souvent trop longs obligent les élèves à utiliser leurs périodes libres du matin, du midi ou du soir pour compléter leurs travaux.

Nous divisons l'année en deux périodes distinctes. La première période s'étend de septembre à janvier et sert à mettre en place des outils de travail (apprentissages de base, mise en place d'outils, programmes, structures). Quant à la deuxième période, elle s'échelonne de février à juin et permet aux étudiantes et étudiants d'utiliser les acquis de la première partie pour entreprendre un projet important (réinvestissement).

Ne pouvant dissocier planification de l'enseignement et planification de l'évaluation, nous reviendrons sur les grandes étapes de planification

MODULE 5

L'ÉVALUATION

OBJECTIF TERMINAL

Planification de l'évaluation

OBJECTIFS INTERMÉDIAIRES

- 5.1 Connaître ce qu'il faut évaluer
- 5.2 Déterminer comment l'évaluer
- 5.3 Construire des instruments d'évaluation

CONTENU

- A. Textes pertinents à l'évaluation
- B. Syllabus de cours
- C. Grilles d'évaluation
- D. Examen-synthèse

DURÉE SUGGÉRÉE: 1 jour

INSTRUMENTATION

A. ACTIVITÉ D'APPRENTISSAGE

- Planification de l'évaluation
- Élaboration d'instruments de mesure
 - o grilles d'observation et d'évaluation
 - o tables de spécifications
 - o examen-synthèse

B. MATRICES DE TRANSPARENTS

- Figures encadrées susceptibles d'être reproduites à des fins pédagogiques

C. MATÉRIEL DIDACTIQUE

- Programme d'études: Introduction à la Science de l'Informatique
- Documents sur l'évaluation pédagogique

D. DOCUMENTATION COMPLÉMENTAIRE

- Textes plus généraux sur l'évaluation formative et sommative disponibles au niveau de la commission scolaire

L'ÉVALUATION

5.1 Introduction

L'évaluation guide l'acquisition des comportements que nous souhaitons voir chez l'étudiant.

L'évaluation fournit aux différents intervenants une information claire et précise sur la progression de l'étudiant dans la poursuite des objectifs.

L'évaluation pédagogique vise l'évaluation des objectifs d'apprentissages définis dans le programme d'études I.S.I.

Bien que nous ayons divisé en 2 modules (module 4 et module 5) la planification et l'évaluation nous pensons que les deux thèmes sont indissociables. En effet, parler de planification sous-entend planification de l'enseignement et planification de l'évaluation.

La planification portera donc sur les points suivants:

1. Déterminer les grands thèmes associés à chaque période de l'année
ex.: Afficher du texte
2. Déterminer les activités de base:
ex.: afficher un texte de présentation
afficher sur 40 colonnes, sur 80 colonnes
afficher plusieurs pages à la suite les unes des autres.
3. Déterminer les exercices d'exploration-découverte et d'acquisitions d'habiletés.
Matériel nécessaire : Disquettes de projets-étudiants
ex.: charger le projet P50 sur la disquette #15 et le faire exécuter.
 - 1) Si vous remplacez la ligne- par la ligne - et faites l'exécution, que se passe-t-il ?
 - 2) A quoi sert la ligne - ?
 - 3) Modifier le programme pour que nous obtenions -
4. Déterminer le mini-projet (Projet personnel ou suggéré ayant pour sujet le thème de l'étape)
ex.: Préparer une page de présentation.

5.2 Les sortes

5.2.1 L'évaluation prédictive AVANT

Déterminer le degré de difficulté
Déterminer les pré-requis et les co-requis

5.2.2 L'évaluation formative PENDANT

Action immédiate auprès de l'élève lors de l'élaboration des mini-projets

Action qui vise à assurer une progression constante des apprentissages

5.2.3 L'évaluation sommative APRÈS

Jugement sur le degré de réalisation des apprentissages

Jugement relatif à la réussite du programme d'études.
Utilisation de grilles d'évaluation.

5.3 Les fonctions

Les différentes formes d'évaluation ont pour but d'aider l'élève dans sa démarche en lui faisant prendre conscience de ce qu'on attend de lui, où il se situe par rapport à cette attente et quels sont les moyens à sa disposition pour réussir à progresser. En fait, nous voulons d'abord et avant tout développer chez l'étudiante et l'étudiant une façon de travailler au lieu d'évaluer de façon ponctuelle l'acquisition de connaissances.

5.4 Buts du cours I.S.I.

5.4.1 Développer les capacités d'adaptation aux changements technologiques

- 5.4.1.1 Connaissance du matériel
- 5.4.1.2 ...
- 5.4.1.3 ...

5.4.2 Développer l'autonomie intellectuelle i.e. développer, chez l'étudiant une pensée logique et structurée.

- 5.4.2.1 L'élève est capable de trouver des stratégies de résolution de problèmes
- 5.4.2.2 L'élève est capable de chercher l'information dont il a besoin
- 5.4.2.3 L'élève est capable d'organiser cette information pour la rendre utilisable
- 5.4.2.4 L'élève est capable de communiquer avec les autres
- 5.4.2.5 L'élève est capable de création

5.5 Les objectifs pédagogiques

5.5.1 Les comportements

Les comportements sont définis par les objectifs du programme.

5.5.2 Aspects opérationnels des objectifs

5.5.2.1 COMMUNICATION: l'objectif nous indique clairement de quoi l'on parle

5.5.2.2 MESURE: l'objectif va nous aider à déterminer dans quelle mesure, dans quel degré un savoir est acquis

5.5.2.3 GUIDAGE: l'objectif nous amène à produire un ou des comportements définis

5.6 Activités privilégiées

5.6.1 Les exercices: activités régies par le professeur, micro-graduées et collées aux objectifs terminaux du programme

exemple: objectif 3.7 Utiliser les instructions d'entrée dans un programme

3.7.1

3.7.2

3.7.3

3.7.4

3.13

5.6.2 Les problèmes: activités réglées par le professeur et regroupant un ensemble de difficultés

exemple: calculer la consommation d'essence

5.6.3 Les projets: activités que l'élève se donne, problème personnel auquel il veut apporter une solution

exemple: dessiner une fusée qui s'élève dans le ciel

5.7 Planification de l'évaluation

5.7.1 Identifier les composantes du programme

5.7.1.1 Classifier les objectifs du programme

- objectifs de 4e secondaire
- objectifs de 4e et 5e secondaire
- objectifs de 5e secondaire

5.7.1.2 Répartir les objectifs dans le temps

5.7.2 Principes directeurs

5.7.2.1 Au début de chaque nouvelle session, lire et expliciter les objectifs poursuivis

5.7.2.2 Donner aux étudiantes et étudiants des moyens d'auto-évaluation

5.7.2.3 Produire des grilles d'observation

5.7.2.4 Produire des grilles d'atteinte des objectifs

5.7.2.5 Produire des grilles se rapportant à la démarche globale du projet

5.0 Orientation possible d'une étape.

Objectif général: Développer, chez l'étudiante et l'étudiant une pensée logique et structurée.

PLANIFICATION

Connaissances et habiletés de base
Développer des outils

SEPT. - JANVIER

Faire des minis-projets

Thème : Apprentissage de base et afficher un texte

Déterminer: - les activités
- les exercices
- le mini-projet

Thème : Faire du graphisme

Déterminer: - les activités
- les exercices
- le mini-projet

Réinvestissement des acquis
Utilisation des outils

FÉVR. - JUIN

Faire un projet de longue haleine

Thème : Faire des calculs

Déterminer: - les activités
- les exercices
- le mini-projet

Thème : Faire de la musique

Déterminer: - les activités
- les exercices
- le mini-projet

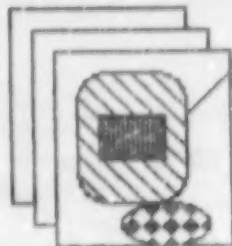
APPRENTISSAGE de base.

Programmation structurée
(structures séquentielles, répétitives
et branchements conditionnels)

PROJET-TEXTE



PROJET-GRAPHIQUE



PROJET-CALCUL



PROJET-MUSIQUE



Septembre à janvier

RÉINVESTISSEMENT

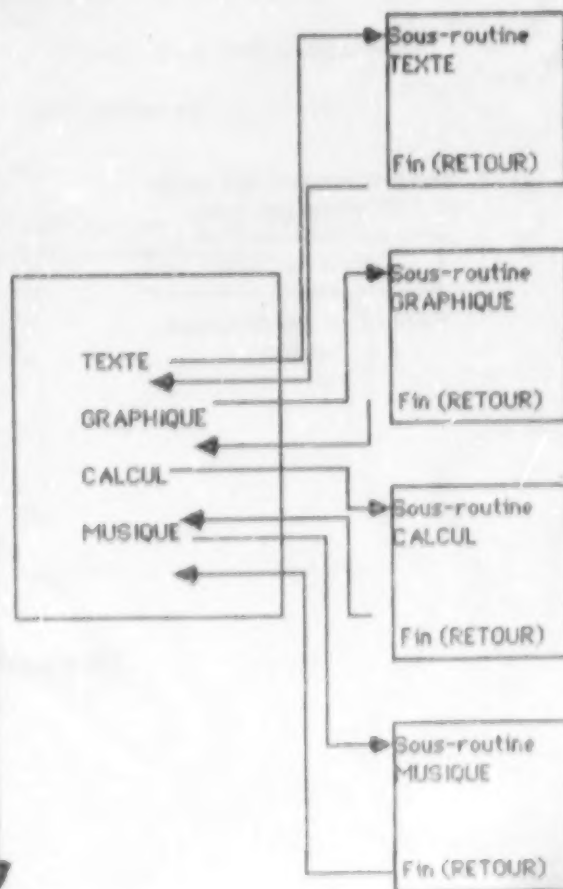
Programmation modulaire
(Utilisation de sous-routines)

Début

Appeler TEXTE
Appeler GRAPHIQUE
Appeler CALCUL
Appeler MUSIQUE

Fin

Février à juin



Objectif général : Développer chez l'étudiante et l'étudiant une pensée logique et structurée.

EVALUATION

Connaissances et habiletés
de base
Développer des outils

SEPT. - JANVIER

Faire des minis-projets

Répartition :

Exercices : 30 % mini-projet 70 %

Évaluation :

Traditionnelle grille d'évaluation

Répartition :

Exercices : 30 % mini-projet 70 %

Évaluation :

Traditionnelle grille d'évaluation

Réinvestissement des acquis
Utilisation des outils

FEVR. - JUIN

Faire un projet de longue
 haleine

Répartition :

Exercices : 30 % mini-projet 70 %

Évaluation :

Traditionnelle grille d'évaluation

Répartition :

Exercices : 30 % mini-projet 70 %

Évaluation :

Traditionnelle grille d'évaluation

Fin d'année :

60 % pour les 4 bulletins + 40 % pour l'examen de fin d'année

(70% projet et 30 % examen objectif)

5.9 Préparer un syllabus

5.9.1 Un plan

5.9.1.1 Un guide ouvert aux changements dans les moyens et dans la répartition

5.9.1.2 Une prise de position vis-à-vis des élèves

5.9.2 Contenu du plan

5.9.2.1 Les objectifs généraux du cours

5.9.2.2 Les objectifs spécifiques du cours

5.9.2.3 Les moyens pour atteindre ces objectifs

5.9.2.4 Un résumé succinct de l'approche privilégiée

5.9.2.5 La répartition chronologique de la matière

5.9.2.6 Le mode d'évaluation

5.9.2.6.1 L'acquisition des connaissances

5.9.2.6.2 L'acquisition d'habiletés

5.10 Évaluation des apprentissages

5.10.1 L'évaluation formative

5.10.1.1 Le nombre de travaux

5.10.1.2 La fréquence des travaux et autres formes d'évaluation

5.10.2 L'évaluation sommative

5.10.2.1 Le nombre et la fréquence d'examens-synthèse et des travaux à long terme

5.10.2.2 Les dates sont inscrites à l'agenda de l'étudiant

5.10.3 La note du bulletin

5.10.3.1 L'évaluation sommative: examens, synthèses, contrôles d'objectifs terminaux et travaux de longue haleine. 70 %

5.10.3.2 L'évaluation formative contrôles, tests 30 %

5.10.3.3 Application de la politique d'évaluation

Informar les étudiants du mode d'évaluation et des objectifs poursuivis au début de chaque période.

5.10.3.4 L'approche par objectif

5.10.3.4.1 Les objectifs 3, 4 et 5 sont évalués par un examen traditionnel

5.10.3.4.2 Les objectifs 1, 2 et 6 nous amènent à évaluer l'acquisition d'attitudes, l'acquisition d'habiletés ainsi que l'appropriation d'une méthode de travail. Il est donc nécessaire de créer des instruments d'évaluation tels que: grilles d'évaluation des acquisitions, grilles d'observation des phases intermédiaires entre le problème posé et sa solution définitive.

5.11 Un exemple de syllabus

NOM DU PROFESSEUR: _____

TITRE DU COURS: Initiation à la science de l'informatique

CODE: 111-432

5.11.1 Planification de l'enseignement

5.11.1.1 Objectif général du cours:

5.11.1.2 Objectifs spécifiques de chaque étape:

étape 1: _____

étape 2: _____

étape 3: _____

étape 4: _____

5.11.1.3 Moyens privilégiés:

5.11.2 Planification de l'évaluation

5.11.2.1 Évaluation sommative

70 %

Nombre/étape / Échéance

5.11.2.1.1 Examens synthèses
sur les objectifs
terminaux

--

5.11.2.1.2 Travaux à long
terme

--

5.11.2.1.3 Autres

--

5.11.2.2 Évaluation formative

30 %

Nombre/étape / Échéance

5.11.2.2.1 Travaux pratiques

--

5.11.2.2.2 Tests

--

5.11.2.2.3 Autres

--

5.12 Instruments de mesure

5.12.1 Feuilles de route pour l'étudiant

Il est intéressant que l'étudiant puisse avoir en sa possession la liste des objectifs poursuivis pendant la période. Personnellement il coche au fur et à mesure ce qu'il identifie être l'atteinte ou la 100% atteinte des objectifs.

5.12.2 Utilisation de l'agenda-étudiant

Bien faire indiquer, par l'étudiant et l'étudiante, les dates importantes à l'agenda.

Examens

Remise d'étapes de projet

Exercices etc.

5.12.3 Grilles d'atteinte des objectifs

Les étapes d'élaboration de projets, telles que présentées dans le présent guide, couvrent les objectifs 1,2 et 3 du programme d'études. L'atteinte de tous les objectifs intermédiaires nous permet de conclure que tel objectif a été atteint à 100% . Cette situation idéale, rarement atteinte, ne saurait être complétée à l'intérieur d'une seule période de l'année. Certains objectifs intermédiaires feront l'objet des acquisitions de première période, d'autres de la deuxième période et ainsi de suite . Ce n'est qu'à la fin de l'année qu'on pourra parler d'atteinte à 95% et plus d'un objectif terminal.

5.12.4 Grille d'évaluation du projet

Démarche globale du projet

Faire prendre conscience de l'écart entre la démarche planifiée et la démarche réelle et pouvoir l'expliquer.

5.12.4.1 Résumé du projet en langage courant

- clarté
- précision

5.12.4.2 Plan du projet

- tâches à réaliser
- lien entre ces différentes tâches

5.12.4.3 Échéancier

- concordance RÉALITÉ VS ÉCHÉANCIER

5.12.4.4 Formulation des tâches en pseudo-langage

5.12.4.5 Codification

5.12.4.6 Documentation

- documentation-programme
- documentation-utilisateur

5.12.4.7 Lisibilité

- renforcement des énoncés
- choix des variables
- blocs...

5.12.4.8 Originalité

5.12.4.9 Progrès de l'élève

5.13 Planification de l'enseignement et de l'évaluation

Les étapes qui suivent vont vous servir de document de base pour entreprendre votre planification.

PAGE 1: Déterminez les thèmes que vous voulez développer pour chaque période de l'année.

PAGE 2: Déterminer: - les activités de base
- les exercices et problèmes à résoudre
- les mini-projets

PAGE 3: Cette page est un aide-mémoire qui comporte la liste des objectifs terminaux se rapportant aux étapes d'élaboration de projets.

PAGE 4: La page 4 donne une liste des grands thèmes de l'année. A vous de compléter et de modifier les activités de base qui sont mentionnées pour chacune.

PAGE 5 et 6 : Gabarit pour une grille d'évaluation. Il est préférable d'élaborer une grille d'évaluation différente pour chacun des mini-projets. Cette grille tiendra compte des objectifs visés par les activités de la période. Référez-vous à la grille proposée à la fin du document, se rapportant au projet de longue haleine.

N.B. La partie de droite vous permet d'indiquer les objectifs terminaux et intermédiaires visés par la planification de l'enseignement.

5.14 Nouveaux concepts

5.14.1 Spécifications

Précisions sur le contenu d'un programme, d'un objectif, d'une épreuve ou d'un item pour des fins d'évaluation.

5.14.2 Table des spécifications

Présentation ordonnée de l'ensemble des notions, visées par un instrument de mesure ou un programme, qui inclut des indications sur le niveau taxonomique de l'apprentissage relié à chaque notion et sur l'importance relative d'une notion ou d'un sous-ensemble de notions par rapport à l'ensemble total.

5.14.3 Examen-synthèse

Épreuve que l'on fait subir à un candidat dans le but d'évaluer ses apprentissages, en vue de la sanction des études.

5.15 Exemple d'une table de spécifications

La répartition de la note de passage se répartit comme suit:

- 60 % pour les quatre (4) examens de l'année
- 40 % pour l'examen final de juin

L'examen final de juin comporte deux volets:

- projet de longue haleine réalisé par les étudiants 70 %

Une grille d'évaluation est préparée par le groupe d'experts et sera envoyée aux écoles.

Objectif général du cours :

THÈMES DE L'ANNÉE

Objectifs terminaux et intermédiaires

[illegible]

Appropriation de l'appareil historique ...	4.1 4.2 4.3 4.4 4.5 4.8 3.13
Énoncé du projet	1.1
Définition du projet	1.2 1.3
Analyse	1.4 2.1 2.5 2.8 2.11 2.12 2.13 2.14 2.15 2.16 2.17 2.20
Réalisation	1.6 1.7 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 3.10 3.11 3.12 3.14 3.15
Tests	1.8
Entretien	1.8
Compte-rendu final	1.9

GRANDS THÈMES: Apprentissage de base

Graphisme:

- écran-texte
- écran-graphique (moyenne et haute résolution)
- traitement d'image
- animation (boucle de temporisation et boucle d'animation)

Affichage de textes:

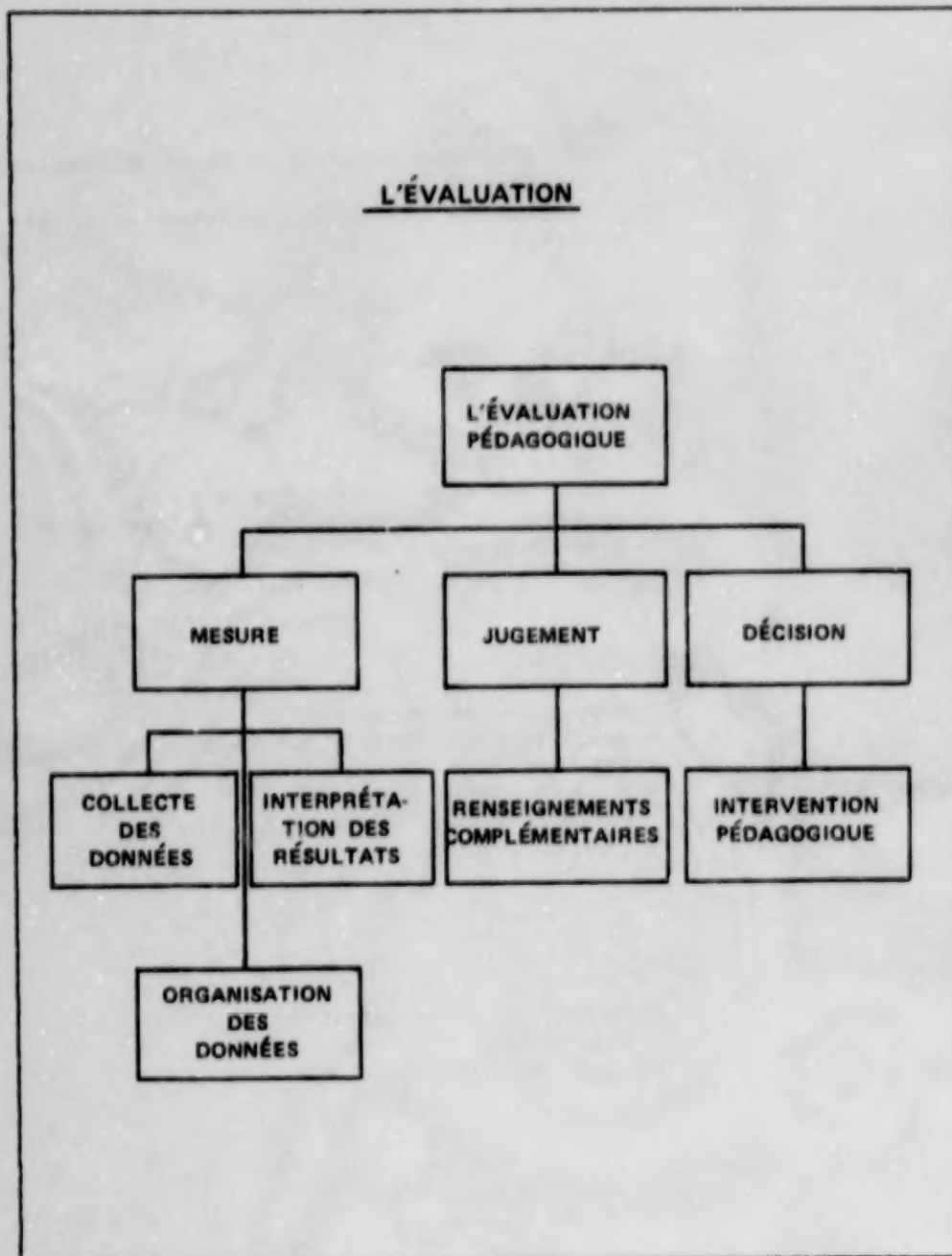
- 40 colonnes
- 80 colonnes
- mise en page
- présentation de plusieurs pages
(ex.: appuyez sur une touche pour continuer)
- opérateurs alphanumériques

Calculs:

- Constantes et variables
- opérateurs booléens et arithmétiques
- affectation
- fonctions prédéfinies
- formules
- fichier interne

Musique:

- traduction d'une partition musicale
- sons divers
- variables alphanumériques
- ...



ÉVALUATION FIG. 1

DÉMARCHE ÉVALUATIVE

1. Mise en situation

Centre d'intérêt de l'activité

- la réceptivité (indices, instruments et hypothèses)

Problème à résoudre

- la compréhension du problème (indices, instrument, hypothèse)
- la participation au problème (indices, instrument, hypothèse)

2. Vécu de l'activité

Exploration de solutions

- la capacité à classer et ordonner des essais de solutions (indices, instrument, hypothèse)
- l'acceptation de soumettre sa démarche à ses camarades (indices, instrument, hypothèse)

Découverte de solutions

- la généralisation de règles de résolution de problèmes

3. Retour sur l'activité

- les étudiants(es) parlent de leur expérience
- les étudiants(es) se questionnent sur leur participation
- les étudiants(es) racontent leurs difficultés et leurs découvertes

PLANIFICATION DE L'ÉVALUATION

Le professeur	L'étudiant
- o connaissance des éléments du programme que les élèves devraient maîtriser - o situations d'apprentissage et tâches d'évaluation de façon à suivre la progression de l'élève. - activités: - de renforcement - d'enrichissement - correctrices - o préparation d'instruments de mesure susceptibles de recueillir les informations nécessaires pour faire le bilan des acquis	- o maîtrise ? - o difficultés d'apprentissage ? - o réussite pour chacun des instruments ?

Voici un exemple de grille d'évaluation:

GROUPE : _____ NOMS : _____ _____ TITRE DU PROJET : _____ DATE DU DEBUT : _____ DATE DE REMISE : _____		E	T	B	M	V	N
		Excellente maîtrise, réussit très facilement.	Très bonne maîtrise, réussit facilement.	Bonne maîtrise, réussit assez bien seul.	Maîtrise, réussit avec de l'aide.	En voie de maîtrise, réussit difficilement.	Non maîtrisé, Ne réussit pas.
1.1 CONSTRUIRE un énoncé clair d'un projet issu d'expérience du quotidien							
1.1.1 LISTER les sujets qui intéressent chaque élève;							
1.1.2 CONSTRUIRE une liste de suggestions de projets avec les autres élèves;							
1.1.3. CHOISIR un projet;							
1.1.4 COMMUNIQUER ses intentions aux autres élèves et au professeur;							
1.1.5. EXPLICITER les différentes étapes du projet à réaliser;							
1.1.6. FAIRE des suggestions pour améliorer les projets de ses camarades;							
1.1.7 INDIQUER les suggestions retenues;							
1.1.8 RÉDIGER un avant-projet.							

5.12 Nouveaux concepts

5.12.1 Spécifications

Précisions sur le contenu d'un programme, d'un objectif, d'une épreuve ou d'un item pour des fins d'évaluation.

5.12.2 Table des spécifications

Présentation ordonnée de l'ensemble des notions visées par un instrument de mesure ou un programme, qui inclut des indications sur le niveau taxonomique de l'apprentissage relié à chaque notion et sur l'importance relative d'une notion ou d'un sous-ensemble de notions par rapport à l'ensemble total.

5.12.3 Examen-synthèse

Épreuve que l'on fait subir à un candidat dans le but d'évaluer ses apprentissages, en vue de la sanction des études.

5.13 Exemple d'une table de spécifications

La répartition de la note de passage se répartit comme suit:

- 60 % pour les quatre (4) examens de l'année
- 40 % pour l'examen final de juin

L'examen final de juin comporte deux volets:

- projet de longue haleine réalisé par les étudiants 70 %

Une grille d'évaluation est préparée par le groupe d'experts et sera envoyée aux écoles.

- examen de juin (évaluation sommative) 30 %

Le contenu de l'examen comporte 30 questions objectives.

La durée sera de deux (2) heures et les étudiants n'ont droit à aucune documentation.

Voici, à titre d'information, la table des spécifications selon les objectifs généraux et intermédiaires du programme d'I.S.I. (c.f. programme d'études au secondaire du MEQ "Introduction à la Science de l'Informatique" p.23 et sq.)

TABLE DES SPÉCIFICATIONS

Examen de quatrième secondaire. (111-432)

Objectifs (1 à 6)	Examen 30 %		Projet 70 %
	Connaissances	Compréhension	Applications
1. Construire des algorithmes	3	6	25
2. Exercer des habiletés	2	7	20
3. Programmer l'ordinateur	0	8	25
4. Connaître le fonctionnement de l'ordinateur	2	0	0
5. Évaluer la place et le rôle de l'ordinateur	1	ou 1	0
6. Développer des attitudes critiques	1	ou 1	0

30 questions

PRÉCISIONS SUR LES OBJECTIFS

1. CONSTRUIRE des algorithmes de programmation afin de résoudre des problèmes relatifs à des expériences du quotidien (9 questions)
 - énumérer dans un ordre logique les actions à faire exécuter par l'ordinateur (1.3.7)
 - représenter sous forme graphique le déroulement logique des étapes du problème ou du projet (1.4.7)
 - traduire certaines étapes de l'algorithme en lecture, écriture affectation, ... (1.5.1)
 - traduire l'algorithme ou une partie de l'algorithme en séquence d'actions (1.5.2)
 - corriger les erreurs de sémantique (1.8.2)
 - corriger les erreurs de logique (1.8.3)
 - tester ou simuler le programme avec quelques données pertinentes (1.8.4)
2. EXERCER des habiletés de résolution de problèmes par une grande variété de stratégies et/ou d'heuristiques (9 questions)
 - partager un problème en sous-problèmes complémentaires et formant une séquence (2.5.2)
 - décrire en langage courant un problème, une solution ou une action (2.11.1)
 - construire un graphe (2.11.4)

- faire le pas-à-pas d'un programme (2.13.2)
- obtenir à chaque étape importante l'état des variables d'un programme (2.13.4)
- vérifier une solution, une donnée, une formule par substitution de valeurs quelconques (2.16.1 à 2.16.3)

3. PROGRAMMER l'ordinateur (8 questions)

- utiliser des constantes numériques (3.1.1)
- utiliser des constantes alphanumériques (3.1.4)
- identifier les symboles employés pour les opérations d'addition, de soustraction, de multiplication, de division et d'exponentiation (3.3.2)
- appliquer les règles de priorité d'exécution des opérations (3.3.3)
- utiliser des instructions d'affectation avec des expressions arithmétiques ou alphanumériques (3.5.2)
- utiliser des instructions d'affectation employant un compteur avec diverses incrémentations (3.5.3)
- identifier les instructions de branchement (3.8.1)
- utiliser les structures itératives de type "répéter... jusqu'à ce que" (3.10.1)
- utiliser les structures itératives de type "tant que" (3.10.2)
- utiliser un compteur pour le contrôle d'une itération (3.10.3)
- utiliser des sous-programmes de type procédures (3.11.5)

4. CONNAÎTRE le fonctionnement matériel et l'histoire de l'ordinateur (2 questions)

- identifier les composantes principales d'un ordinateur: l'unité centrale, les unités périphériques (4.1.1)
- identifier les opérations de base d'un ordinateur (4.1.2)
- décrire sommairement la fonction des composantes principales d'un ordinateur (4.1.3)
- distinguer les classes d'unités périphériques: les unités d'entrée, les unités de sortie, les unités de mémoire auxiliaire (4.3.1)
- identifier et décrire sommairement des unités d'entrée: le clavier alphanumérique, le lecteur de cartes, le lecteur de bandes, etc. (4.3.2)
- identifier et décrire sommairement des unités de sortie: l'écran de visualisation, l'imprimante, le traceur de courbes, etc. (4.3.3)
- distinguer entre un langage de haut niveau et un langage de bas niveau (4.5.1)
- distinguer entre le logiciel et le matériel (4.5.7)
- décrire les débuts de l'ère des ordinateurs, le rôle des premiers inventeurs et identifier les grandes étapes de l'évolution des ordinateurs (4.8.1 à 4.8.3)

5. ÉVALUER la place et le rôle de l'ordinateur dans la société
(1 question)

- décrire sommairement la place de l'ordinateur dans les communications, le système éducatif, ... (5.1.1 à 5.1.7)
- identifier que la plupart des fautes imputées à la machine sont en fait des erreurs humaines (5.2.2)
- identifier des catégories d'emplois créées par l'informatisation et d'autres qui disparaissent (5.4.1 et 5.4.2)

6. DÉVELOPPER des attitudes critiques vis-à-vis du traitement automatique de l'information (l'informatique) (1 question)

- l'élève perçoit l'informatisation comme un élément non menaçant et comme susceptible d'être à son service (6.1.1 et 6.1.2)
- l'élève perçoit la programmation comme une activité de résolution de problèmes qui pose parfois de grands défis (6.3.2)

5.16 Un exemple d'examen-synthèse

Avertissement: L'examen suivant n'est qu'un exemple parmi tant d'autres et ne devrait, en aucun cas, servir de modèle absolu.

E X A M E N

DATE: juin 198
HEURE: 13h30 à 15h00

111-432

INTRODUCTION À LA SCIENCE DE L'INFORMATIQUE(4)

111-432

DIRECTIVES

1. Lire attentivement les instructions qui apparaissent au verso de la feuille réponses:
"Directives à l'intention des candidats".
2. Dans la section 7 de cette feuille de réponses, noircir au crayon à mine HB la case ovale encerclant la lettre correspondant à la réponse choisie.
3. Répondre à toutes les questions de l'examen.
4. L'usage de notes, de dictionnaires ou de manuels est interdit. L'usage de calculatrices est interdit.
5. Remettre le questionnaire au surveillant en même temps que votre feuille de réponses.

N.B.: Ne rien écrire dans les cases des sections 8 et 9 de la feuille de réponses.

INFORMATIQUE 432

PARTIE A

1. Le contenu de la mémoire morte (ROM)

- A) se détruit lorsque l'on éteint l'ordinateur.
- B) se modifie en chargeant le programme.
- C) demeure intact lorsqu'on éteint l'ordinateur.
- D) est le même sur tous les ordinateurs.

2. Le système de base d'un micro-ordinateur se compose

- 1) d'un écran
- 2) d'un clavier
- 3) d'une unité centrale de traitement
- 4) d'une mémoire

- A) De 3 seulement
- B) De 1 - 2 - 3 seulement
- C) De 2 - 3 - 4 seulement
- D) De 1 - 2 - 3 - 4

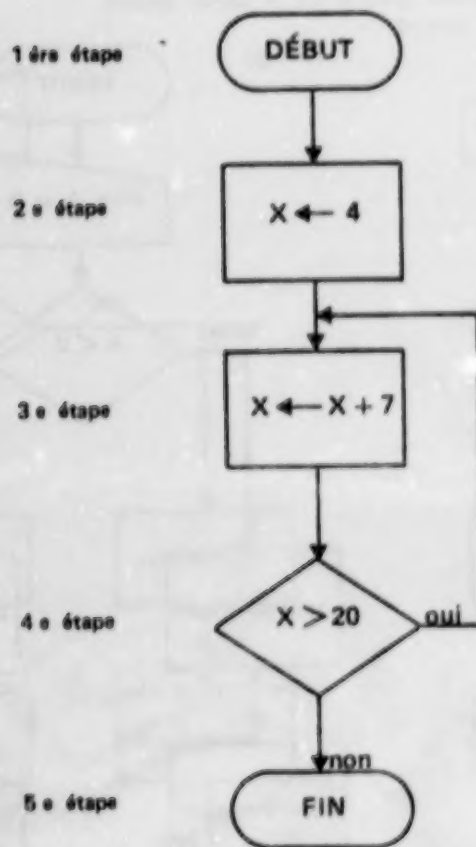
3. Combien d'octets ou de caractères peut-on emmagasiner sur une disquette ordinaire de cinq pouces et un quart?

- A) Entre 100 000 et 1 000 000
- B) Entre 500 et 10 000
- C) Entre 10 000 et 100 000
- D) 500 et moins
- E) 1 000 000 et plus

4. Le programme "BOTTIN" est enregistré sur une disquette. Quelle série de commandes faut-il donner à l'ordinateur pour qu'il l'exécute?
- A) Imprimer et lister
 - B) Lister et exécuter
 - C) Charger en mémoire en lister
 - D) Charger en mémoire et exécuter
 - E) Charger en mémoire et imprimer les instructions
5. D'un ordinateur situé à Montréal, je veux envoyer par ligne téléphonique des données à un autre ordinateur situé à Longueuil. Il faut que les deux ordinateurs soient munis d'un
- A) modulateur
 - B) modem
 - C) démodulateur
 - D) décodeur numérique
 - E) moniteur
6. L'informatique envahit toutes les sphères de l'activité humaine. Laquelle des propositions suivantes est "FAUSSE"?
- A) L'accès "aux banques de données" rend service aux industries.
 - B) "Le traitement de textes" facilite et améliore le travail de secrétariat.
 - C) L'accès "aux banques de données" menace la vie privée des gens.
 - D) La robotique (l'utilisation du robot dans l'industrie pour les tâches répétitives) va amener l'homme à ne plus travailler.
7. Lequel des énoncés suivants est faux?
- A) La programmation est une activité de création.
 - B) Un programme très bien structuré ne peut plus être amélioré.
 - C) La bureautique, c'est l'application de l'informatique aux travaux de bureau.
 - D) Les industries de haute technologie sont des entreprises où l'informatique joue un grand rôle.
 - E) L'ordinateur est une manifestation du génie de l'homme.

PARTIE B

8. Soit l'ordinogramme suivant.



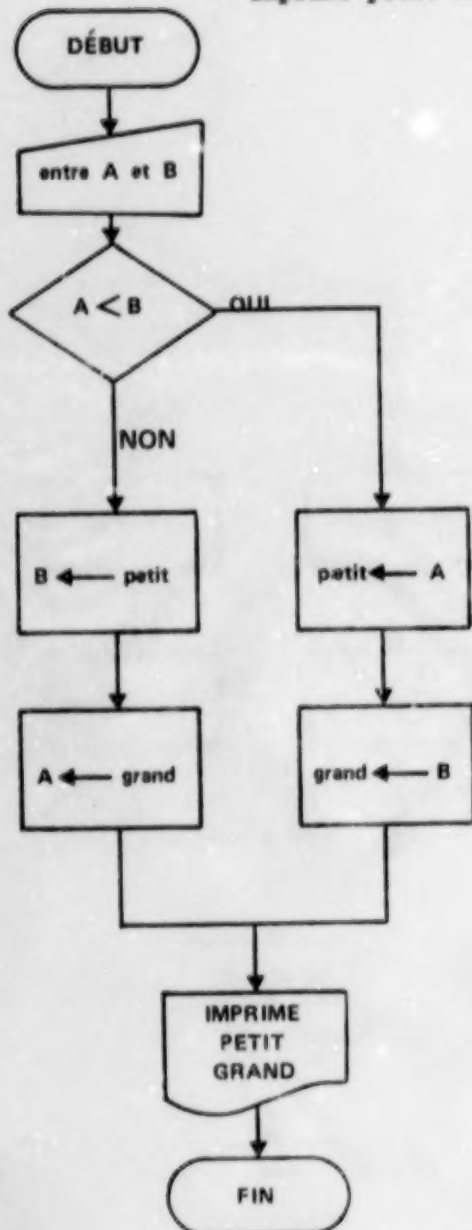
L'étape 4 illustre

- A) un branchement inconditionnel
- B) un sous-programme
- C) un branchement conditionnel
- D) une affectation

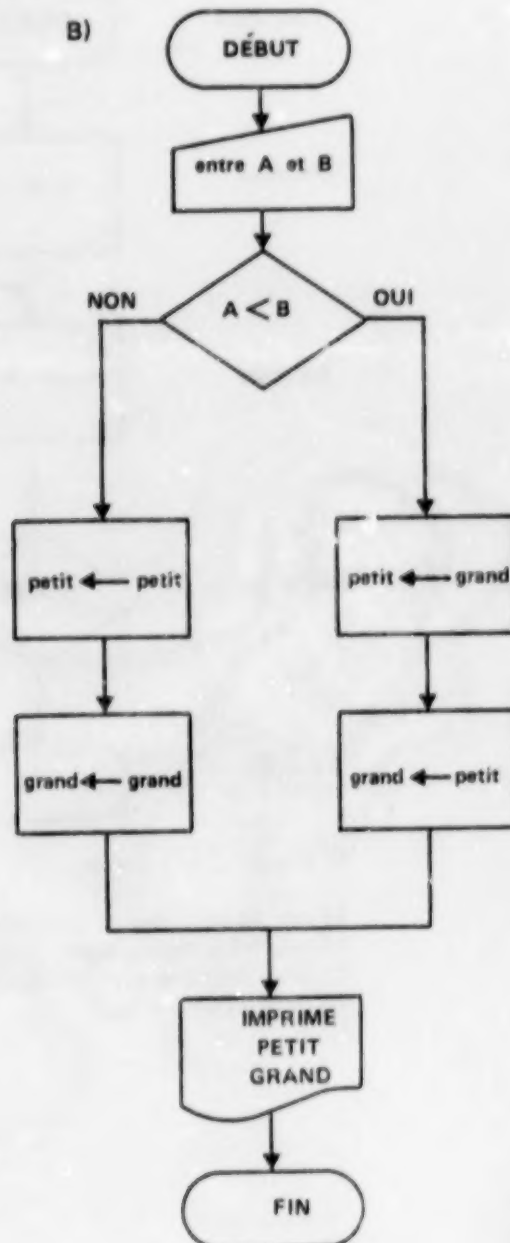
9. Quel ordiogramme correspond à l'algorithme ci-dessous parmi les quatre (4) choix suivants:

- entre 2 nombres: A et B
- est-ce que A est plus petit que B?
- si oui, place A dans petit et B dans grand
- si non, place B dans petit et A dans grand
- imprime petit et grand

A)

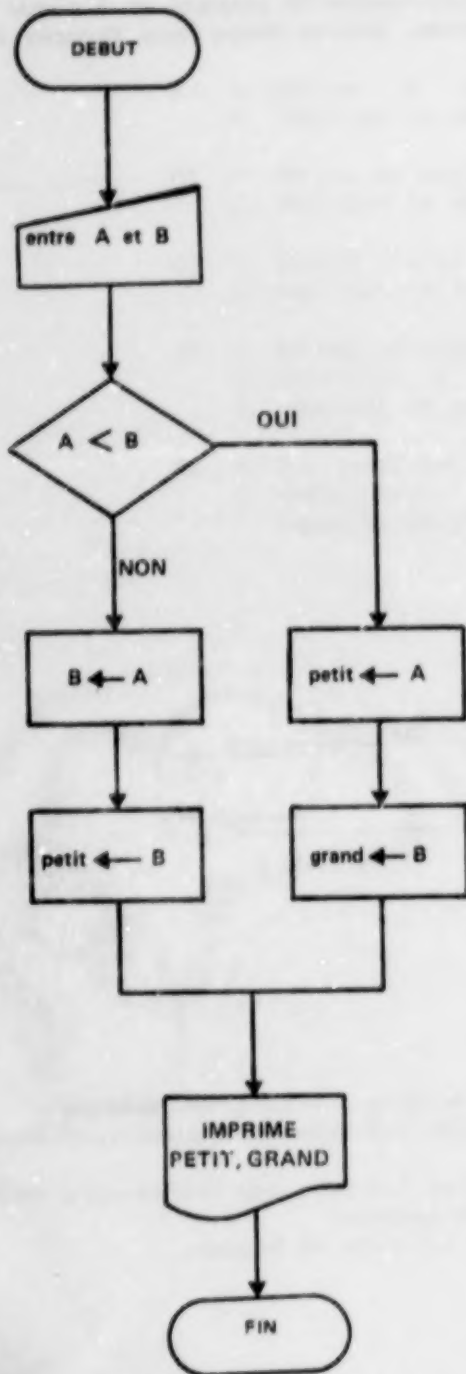


B)

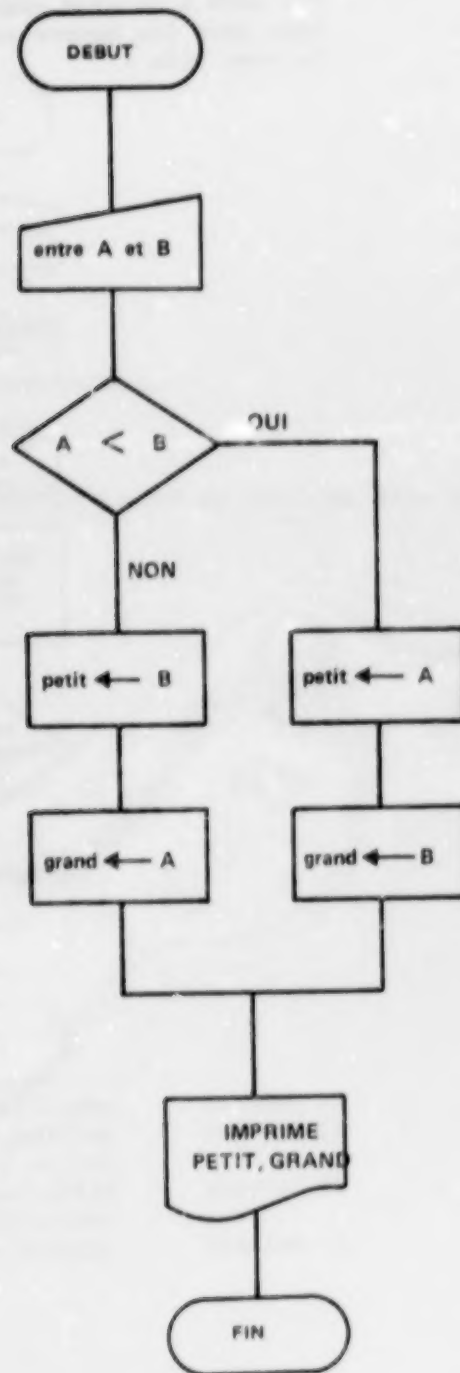


9. - (suite)

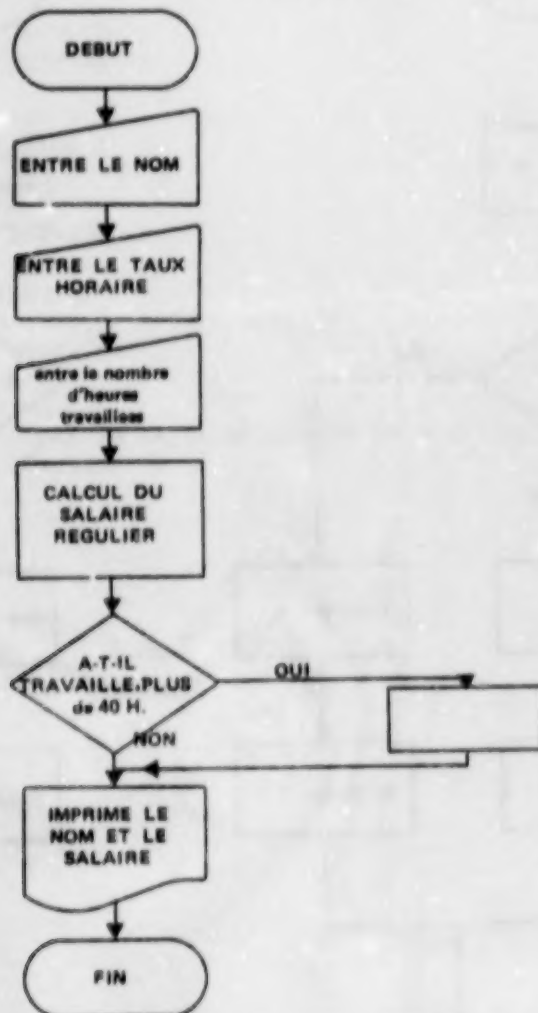
C)



D)



10. Ce programme est conçu pour calculer le salaire d'un employé à partir d'un taux horaire et du nombre d'heures travaillées. Si cet employé est payé à un taux régulier pour 40 heures de travail et à temps et demi pour les heures supplémentaires, quelle étape doit figurer dans la case vide ?



- A) Salaire
- B) Salaire
- C) Salaire
- D) Salaire

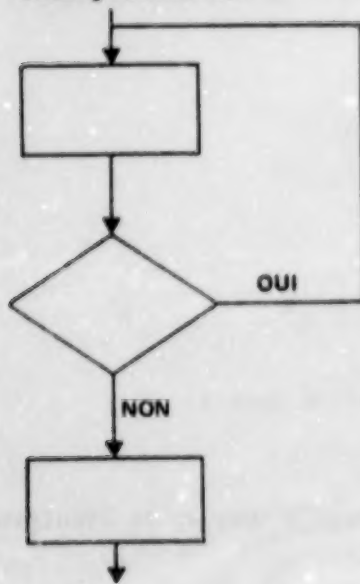
son salaire régulier plus 1,5 fois son salaire.
 salaire régulier plus 1,5 fois ses heures supplémen-
 taires.
 salaire régulier plus 1,5 fois ses heures supplémen-
 taires fois le taux horaire.
 salaire régulier + 1,5 fois 40 heures.

11. On veut calculer et imprimer l'aire de n'importe quel cercle à l'aide de la formule $A = \pi R^2$. Quel serait le bon algorithme si A représente l'aire, R le rayon et π le nombre 3,14?

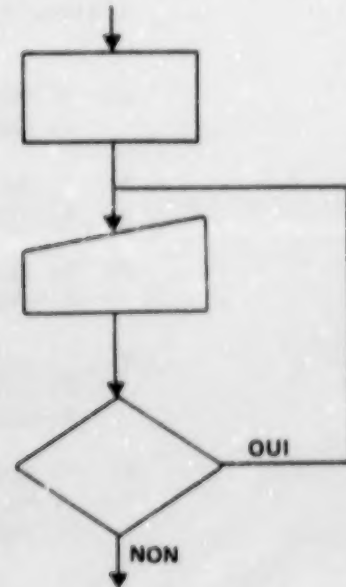
- A) 1. Mettre R dans A.
2. Imprimer le résultat A.
- B) 1. Entrer un rayon R.
2. Imprimer le résultat A.
- C) 1. Entrer l'aire A.
2. Imprimer le résultat A.
- D) 1. Entrer un rayon R.
2. Calculer R^2 et le mettre dans A.
3. Imprimer le résultat A.
- E) 1. Entrer le rayon R.
2. Multiplier par le rayon et placer le résultat dans A.
3. Imprimer le résultat A.

12. Quel ordiogramme représente une sous-routine (sous-programme ou sous-procédure) ?

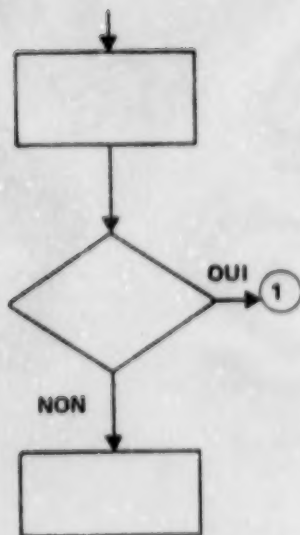
A)



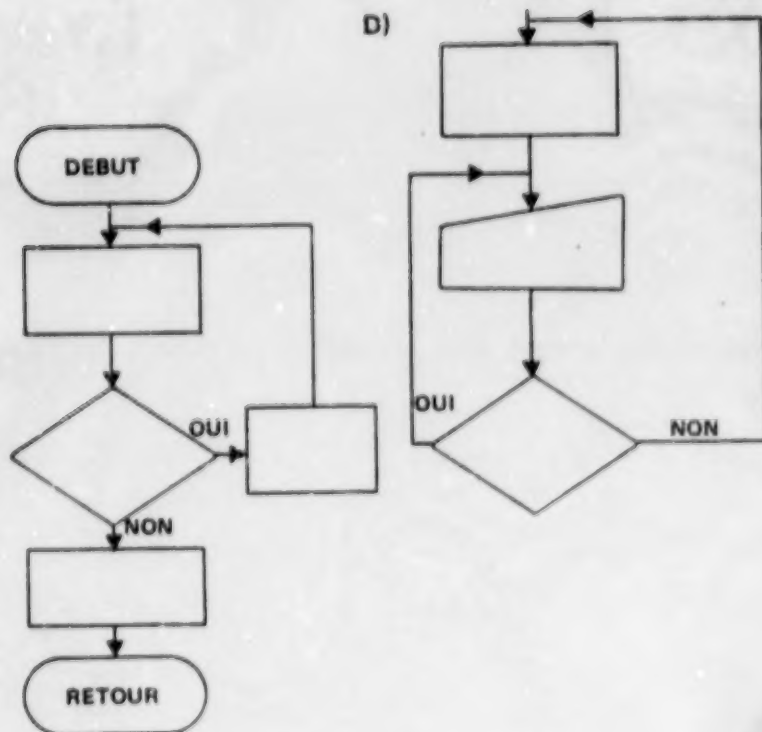
B)



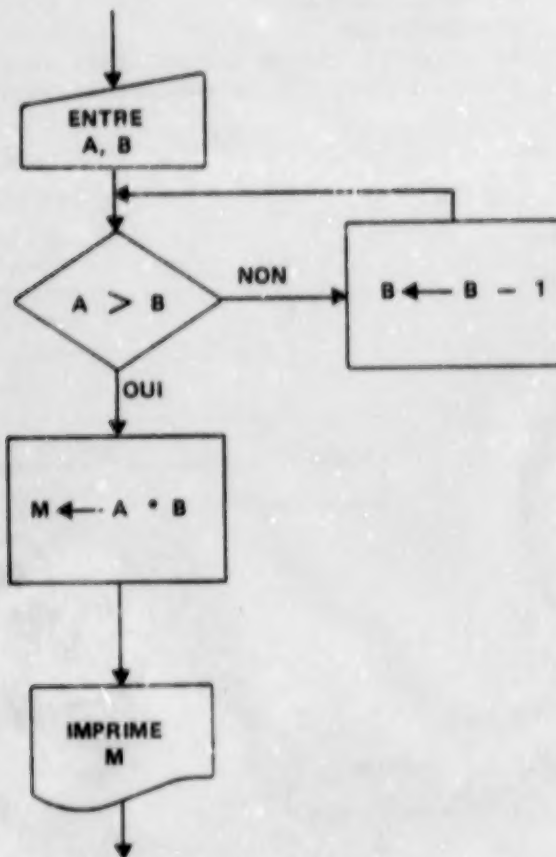
C)



D)



13. Soit l'ordinogramme suivant.

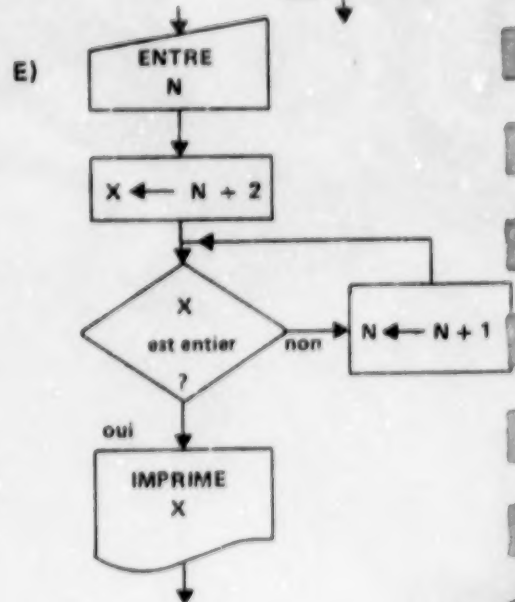
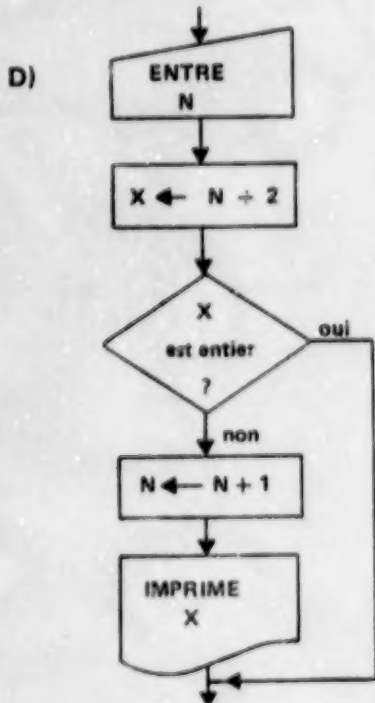
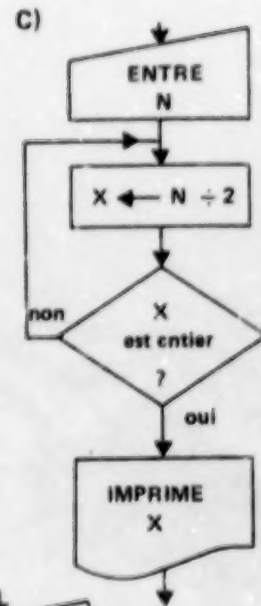
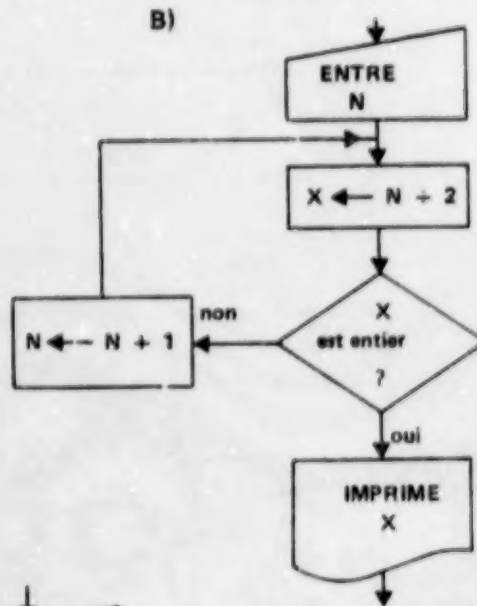
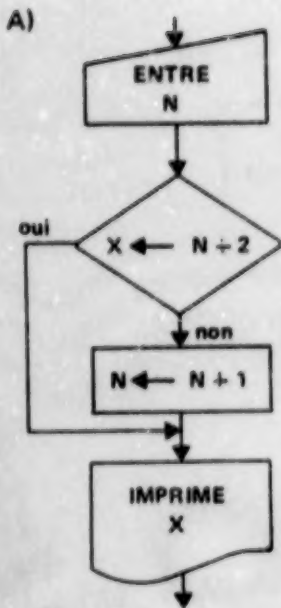


Si on entre 4 pour A et 8 pour B, combien de fois faut-il passer dans la branche du Non?

- A) 2 fois
- B) 3 fois
- C) 4 fois
- D) 5 fois
- E) 8 fois

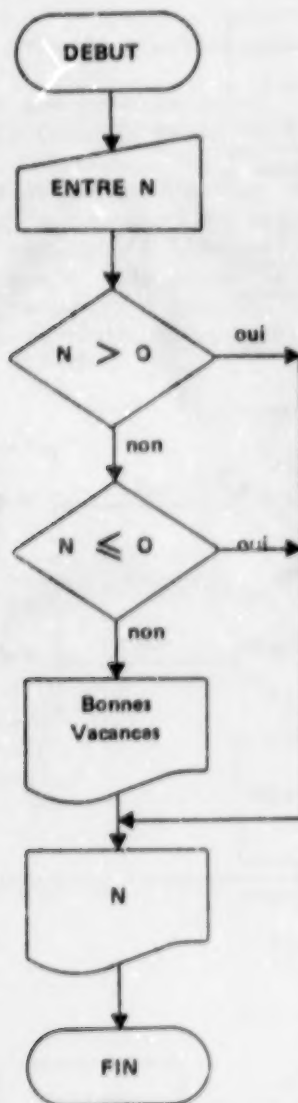
14. Soit l'algorithme suivant:

Entrez un nombre.
Divisez-le par 2.
Si c'est un nombre entier alors imprimez-le.
Sinon, ajoutez 1 à ce nombre et retournez à la division par 2

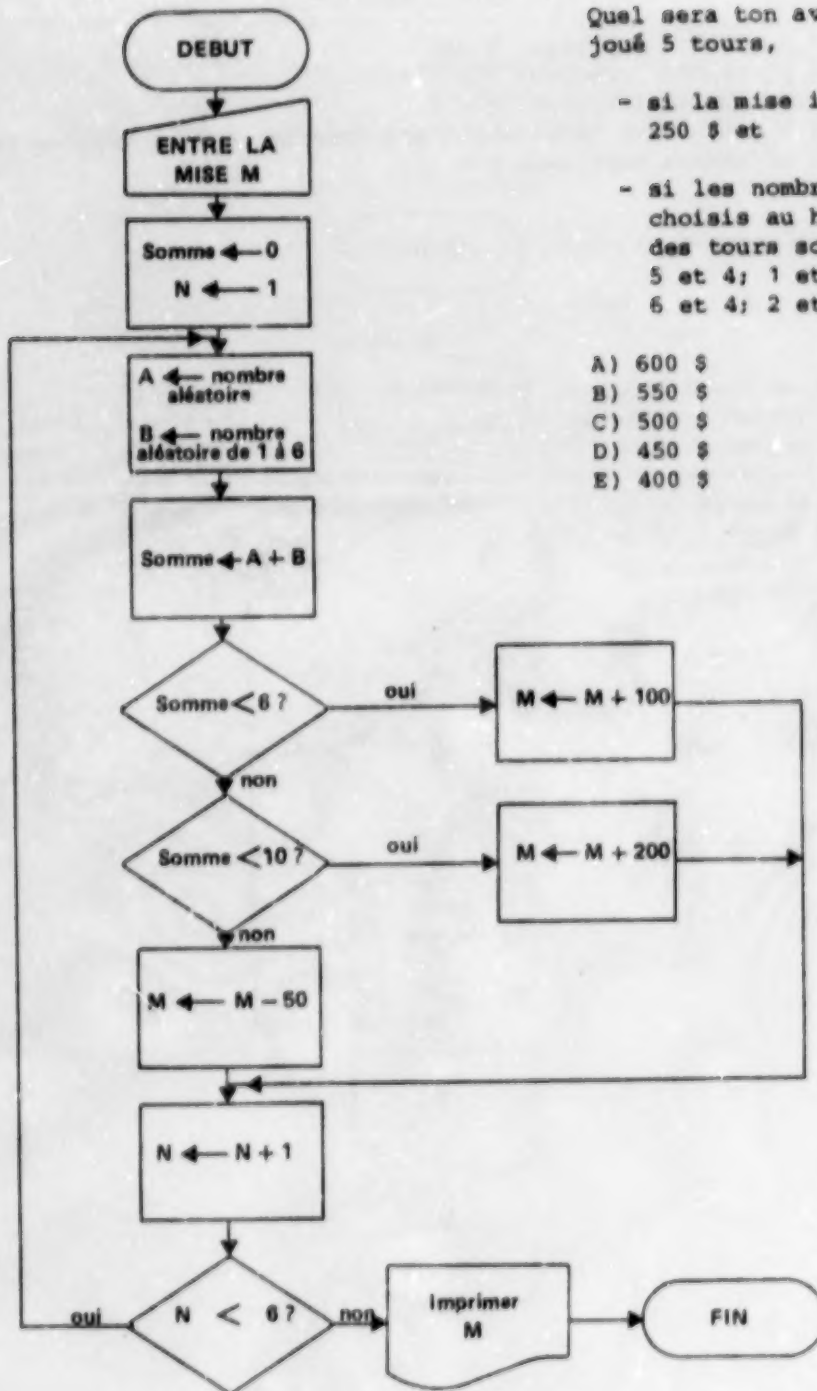


15. Quelle valeur faut-il attribuer à la variable N si l'on veut faire afficher le message "Bonnes vacances"?

- A) Un nombre supérieur à 100
- B) Un nombre inférieur à 0
- C) Un nombre inférieur à -10
- D) Aucun nombre ne permet d'afficher le message "Bonnes vacances"
- E) Un nombre supérieur à 0



16. Soit l'ordinogramme suivant:



Quel sera ton avoir après avoir joué 5 tours,

- si la mise initiale est de 250 \$ et

- si les nombres (A et B) choisis au hasard pour chacun des tours sont successivement 5 et 4; 1 et 1; 6 et 6; 6 et 4; 2 et 17

- A) 600 \$
- B) 550 \$
- C) 500 \$
- D) 450 \$
- E) 400 \$

PARTIE C

17. Cet algorithme doit permettre de calculer le montant d'une facture pour plusieurs articles, certains taxables, d'autres non taxables. Quelle étape manque à cet algorithme?

- 1) On entre le prix d'un article
 - 2) Est-il taxable?
 - 3) Si oui, trouve le prix de l'article avec taxe et additionnele au total
 - 4) Si non, additionne son prix au total
 - 5) On fait imprimer le montant de la facture
-
- A) On oublie de calculer la taxe sur le montant de la facture.
 - B) On oublie de poser la question suivante: Est-ce le dernier article? Si oui, retourne à l'étape 5, sinon va à l'étape 1.
 - C) On oublie après l'étape 3 de poser la question: Est-ce le dernier article? Si oui, retourne à l'étape 1, sinon va à l'étape 4.
 - D) On oublie de faire le grand total après l'étape 4.

18. Soit le programme suivant: (A\$ désigne une chaîne de caractères).

```
A$ ← "JE BÉGAYE"  
B$ ← "TOUJOURS"  
C$ ← "JE SUIS GUÉRI"  
REPETER  
  Ecrire A$  
  Ecrire B$  
JUSQU'A la rencontre de C$  
Ecrire C$
```

Laquelle des propositions suivantes est VRAIE?

A) L'ordinateur affiche une fois à l'écran la phrase

JE SUIS GUÉRI

B) L'ordinateur affiche sans arrêt à l'écran les phrases suivantes:

JE BÉGAYE
TOUJOURS
JE SUIS GUÉRI

C) L'ordinateur affiche une fois à l'écran les phrases suivantes:

JE BÉGAYE
TOUJOURS
JE SUIS GUÉRI

D) L'ordinateur affiche sans arrêt la phrase suivante:

JE BÉGAYE
TOUJOURS

mais il n'affichera jamais à l'écran la phrase:

JE SUIS GUÉRI

19. Soit le programme suivant: (C'est un compteur)

```
C ← 1
Répéter
  P ← 2 exposant C
  Écrire P
  C ← C + 1
JUSQU'À ce que C = 6
```

Laquelle des propositions suivantes est vraie?

Ce programme permet

- A) d'afficher les 5 premiers nombres naturels pairs
- B) d'afficher 5 fois le caractère "p"
- C) d'afficher les 5 premières puissances entières et positives de 2
- D) de faire la somme des 5 premiers nombres naturels

20. Soit le programme suivant:

```
Entrer Salaire
SI salaire > 20 000
  ALORS impôt ← salaire X 15 %
SINON SI salaire > 10 000
  ALORS impôt ← salaire X 10 %
  SINON impôt ← 0
Écrire "Impôt à payer = ", impôt, "$"
```

Si une personne reçoit un salaire de 14 000 \$, l'impôt à payer est égal à

- A) 2 100 \$
- B) 0 \$
- C) 1 000 \$
- D) 1 400 \$
- E) 3 000 \$

21. Soit la fonction: $y = 2x - 5x + 6$.

Quel est le programme en pseudo-code qui calcule la valeur de y quand x varie de -5 à 5 par pas de $1/2$ et qui imprime les valeurs de x et y .

A) $x \leftarrow -5$
Répète
 $y \leftarrow 2x - 5x + 6$
 écris x et y
 $x \leftarrow x + 0,5$
jusqu'à ce que $x = 5,5$
Fin

B) $x \leftarrow -5$
Répète
 $y \leftarrow 2x - 5x + 6$
 écris x et y
 $y \leftarrow y + 1/2$
jusqu'à ce que $x = 5,5$
Fin

C) $y \leftarrow 2x - 5x + 6$
Répète
 $x \leftarrow -5$
 écris x et y
 $y = 1/2$
jusqu'à ce que $x = 5,5$
Fin

D) $x \leftarrow -5$
Répète
 $y \leftarrow 2x - 5x + 6$
 écris x et y
 $x \leftarrow x + 1/2$
jusqu'à ce que $x = 5$
Fin

E) Répète
 $y \leftarrow -5$
 $y \leftarrow 2x - 5x + 6$
 écris x et y
jusqu'à ce que $x = 5$
Fin

Répondre aux questions 22 et 23 à partir du programme et des sous-programmes ou encore des procédures ci-dessous

Programme PRINCIPAL

```
Écrire "menu"
Écrire "1. Addition de 2 nombres"
Écrire "2. Soustraction de 2 nombres"
Écrire "3. Multiplication de 2 nombres"
Écrire "Entre ton choix";
Entrer C
Appeler CHOIX
Selon C appeler SOMME, DIFFÉRENCE, PRODUIT
Écrire "Veux-tu faire un autre choix (1 = oui, 2 = non)";
Entrer C1
SI C1 = 1
    ALORS retourner au menu
    SINON Écrire un message d'adieu
```

Sous-programme CHOIX

```
Écrire "entre deux nombres";
Entrer A, B
SI C = 1 ALORS R1 ← A + B
SI C = 2 ALORS R1 ← A - B
SI C = 3 ALORS R1 ← A × B
Écrire "entre ta réponse";
Entrer R
RETOUR
```

Sous-programme somme

```
SI R = R1
    ALORS écrire "bravo"
    SINON écrire "tu sommeilles"
RETOUR
```

Répondre aux questions 22 et 23 à partir du programme et des sous-programmes ou encore des procédures ci-dessous (suite)

Sous-programme DIFFÉRENCE

```
SI R = R1
  ALORS écrire "excellent"
  SINON écrire "dommage"
RETOUR
```

Sous-programme PRODUIT

```
SI R = R1
  ALORS écrire "formidable"
  SINON écrire "Ahhhhhh..."
RETOUR
```

22. Si nous nous reportons au programme PRINCIPAL, l'action suivante:

Selon C appeler SOMME, DIFFÉRENCE, PRODUIT représente

- A) une structure répétitive
- B) une structure alternative
- C) une structure séquentielle
- D) une structure récursive

23. Si nous donnons à la variable C la valeur 4, le programme PRINCIPAL fera appel

- A) à 2 sous-programmes
- B) à 1 sous-programme
- C) aux 4 sous-programmes
- D) à aucun des sous-programmes

24. Ce programme calcule la moyenne d'un groupe d'élèves et la moyenne de chacun de ces élèves. Si chaque élève a cumulé 5 notes sur 100, qu'est-ce qui ne fonctionne pas dans ce raisonnement qui fait afficher la moyenne du groupe seulement.

Étape 1 : Somme ←
 moyenne de l'élève ←
 moyenne du groupe ←
 somme des moyennes ←
Étape 2 : Entre le nombre d'étudiants du groupe
Étape 3 : Répète
Étape 4 : Répète
Étape 5 : Entre la note de l'élève sur 100
Étape 6 : Somme ← somme + note de l'élève
Étape 7 : Jusqu'à ce qu'il y ait 5 notes d'entrer
Étape 8 : Moyenne de l'élève ← somme divisée par 5
Étape 9 : Somme des moyennes ← somme des moyennes + moyenne
 de l'élève
Étape 10: Jusqu'au nombre d'étudiants désigné
Étape 11: Moyenne du groupe ← somme des moyennes divisée par
 le nombre d'élèves
Étape 12: Imprime la moyenne du groupe

- A) Après l'étape 8, il faut remettre somme à zéro
B) Après l'étape 4, il faut remettre somme à zéro
C) Après l'étape 9, il faut remettre moyenne de l'élève à zéro
D) Après l'étape 3, il faut remettre la somme des moyennes à zéro
E) Après l'étape 6, il faut remettre la moyenne d'élèves à zéro

25. Supposons que l'ordinateur choisit UN MOT qu'il place dans chaîne 1.
Que simule alors le programme suivant?
(C et D sont des compteurs)

Lire chaîne 1

L ← longueur de chaîne 1

D ← L

Répéter

Entrer une lettre dans chaîne 2

Pour C variant de 1 à L

Si chaîne 2 est la C ième lettre de chaîne 1

ALORS placer chaîne 2 dans la C ième case de chaîne 3

D ← D - 1

SINON écrire "Mauvais choix"

Fin de boucle

JUSQU'À ce que D =

Ce programme...

- A) extrait les voyelles de chaîne 1
- B) découvre le mot choisi par l'ordinateur
- C) détermine si chaîne 1 est un palindrome (un mot qui se lit dans les deux sens)
- D) place chaîne 1 dans l'ordre inverse

PARTIE D

26. Soit le programme suivant:

```
Lire A, B
Q ← 0
RÉPÉTER TANT QUE A > B
    A ← A - B
    Q ← Q + 1
Fin du TANT QUE
R ← A
Écrire le contenu de Q
```

Si l'on attribue aux variables A et B les valeurs 23 et 5, quel sera le contenu de la variable Q?

- A) 3
- B) 5
- C) 4
- D) 6

27. Soit le programme suivant:

```
Lire un nombre N
S ← N
RÉPÉTER TANT QUE N > 1
    N ← N - 1
    S ← S + N
Fin du TANT QUE
Écrire "travail terminé"
```

Si on entre 5 dans la case N, quel sera le contenu de la case S?

- A) 5
- B) 15
- C) 14
- D) 20

28. Soit le programme suivant:

```
Entrer A, B
 $X \leftarrow A + B$ 
SI  $X \geq 10$ 
    ALORS écrire X
    SINON entrer 2 autres nombres
```

Ce programme permet...

- A) d'afficher la somme de 2 nombres si celle-ci est supérieure à 10
- B) d'afficher la somme de 2 nombres quelconques
- C) d'afficher la somme de 2 nombres si seulement ces nombres sont supérieurs à 10
- D) d'afficher la somme de 2 nombres si seulement l'un d'eux est supérieur à 10

29. Soit le programme suivant: (C'est un compteur)

```
C  $\leftarrow$  1
T  $\leftarrow$  0
RÉPÉTER
    Lire un nombre N
    SI  $N \leq 15$  et  $N \leq 40$ 
        ALORS  $T \leftarrow T + N$ 
        SINON  $T \leftarrow T$ 
    C  $\leftarrow$  C + 1
JUSQU'À ce que C = 6
```

Si la variable N prend les valeurs suivantes (5, 12, 18, 32 et 40), quel sera le contenu de la variable T?

- A) 107
- B) 40
- C) 50
- D) 57
- E) 90

30. Soit le programme suivant: (C'est un compteur)

```
N ← 5
Entrer un nombre A
POUR C variant de 1 à N
  Entrer un nombre B
  SI A < B
    ALORS X ← A: A ← B: B ← X
  Fin de boucle
```

Si A = 25 et B prend successivement les valeurs 17, 42, 25, 6 et 18
le contenu de la case A est:

- A) 42
- B) 25
- C) 18
- D) 17
- E) 6

GRILLE D'ÉVALUATION

SE RAPPORTANT AU PROJET DE LONGUE HALEINE

		1	2	3	4	5
①	L'élève a formulé un énoncé clair du projet à réaliser à l'aide d'un scénario ou d'une simulation écrite.					
②	L'élève a produit un plan de travail i.e. il a bien identifié les tâches à accomplir. STRUCTUROGRAMME.					
③	Les éléments de base de l'algorithme sont bien définis i.e. les composantes de base du programme sont bien développées.					
④	L'élève a décrit les étapes de sa démarche en pseudo-code et/ou forme graphique. PSEUDO-LANGAGE et/ou ORDINOGRAMME.					
⑤	Le programme fonctionne bien, techniquement parlant. L'exécution se déroule normalement.					
⑥	Le logiciel couvre tous les cas de traitements normaux, extrêmes et exceptionnels (tous les cas ont été prévus).					
⑦	Des procédures de récupération en cas d'urgence et des messages d'erreurs clairs et descriptifs sont présents.					

1 2 3 4 5

8	Le programme est conforme aux spécifications décrites dans l'analyse et aux transformations envisagées.					
9	Les structures conventionnelles sont présentes (structures séquentielles, sélectives et répétitives).					
10	Documentation-programme (identification des modules DEBUT-FIN, liste des variables, titre, auteurs, type d'appareil...)					
11	Documentation-utilisateur (explications claires et optionnelles, fonction d'aide)					
12	Chaque module remplit bien son rôle					
13	Le style du programme est consistant (un énoncé exécutable par ligne, espacements REM indentation...)					
14	Originalité - innovation - présentation					

/35

☐ / 35 + ☐ / 35 = ☐ / 70

BIBLIOGRAPHIE GÉNÉRALE

1. Robillard, Pierre N. LE LOGICIEL: de sa conception à sa maintenance, GAÉTAN MORIN éditeur, 1985.
2. Gouvernement du Québec, Direction des programmes. Introduction à la Science de l'Informatique, Programmes d'études secondaires, 1982.
3. Université du Québec, Télé-Université. Quelques lignes directrices concernant l'évaluation de projets d'élèves, Marcel Labelle et Alain Taurisson, sept. 1979.
4. Gouvernement du Québec, G.R.M.S. Initiation à l'ordinateur dans l'enseignement général au secondaire. 1979.
5. Gouvernement du Québec, D.G.D.P. Guide pédagogique du cours d'Introduction à la Science de l'Informatique, 1984.
6. Roger W. Haigh/Loren E. Radford. Basic pour MICRO-ORDINATEURS Apple, Pet et TRS-80, traduction Johanne de Luca, Modulo Éditeur, 1983.
7. Jack Emmerichs. APPRENDRE à PROGRAMMER, Modulo Éditeur, 1985.
8. Mike James. MIEUX PROGRAMMER, Modulo Éditeur, 1985.
9. Boris Allan. LOGO, Modulo éditeur, 1984.
10. Douglas Downing et Mark Yushini. PASCAL SANS PEINE, TURGEON, 1985.
11. Chantal et Patrice Richard. Initiation à l'algorithmie, Collection DIA/Éditions BELIN, 1981.

BIBLIOGRAPHIE GÉNÉRALE (suite)

12. John G. Kemeny and Thomas E. Kurtz. BASIC PROGRAMMING, John Wiley and Sons Inc., 1971.
13. Thomas A. Dwyer, Margot Critchfield. COMPUTER RESOURCE Book-Algebra, Houghton Mifflin Company/Boston, 1975.
14. James S. Coan. Advanced basic application and problems, Hayden book company, Inc., 1977.
15. Kan Skier. Top-Down BASIC for the TRS-80 Color Computer, Mc Graw-Hill, 1983.
16. Donald Hearn, M. Pauline Baker. Microcomputer GRAPHICS techniques & applications, Prentice-Hall Inc., 1983.
17. Mitchell Waite & Christopher L. Morgan. GRAPHICS primer for the IBM PC, Mc Graw-Hill, 1983.
18. Milton D. Rosenaw Jr., Marsha D. Lawin. Software Project Management step by step, Lifetime Learning Publications, Ten Davis Drive Belmont, California 94002.
19. David E. Simon. Graphics from the ground-up, Hayden Book Company.
20. Guides d'utilisation
 - Mac Project, Solosoft, Apple Computer Inc., 1984.
 - Harvard Project Manager, Harvard Software Inc., 1984.
21. Y. Auger, G. Barot, P. Leroux. Le Logo en français sur commodore 64, Edition FM/Magnard, 1985.
22. Tony Fabbri, Animation, Games, and Sound for the IBM PC, Prentice-Hall personal computing series, 1983.
23. Claude Coulombe, Richard Young, La programmation structurée en Basic, Turgeon 1986.

ANNEXE 1

OUTILS DE CONCEPTION

L'ORDINOGRAMME

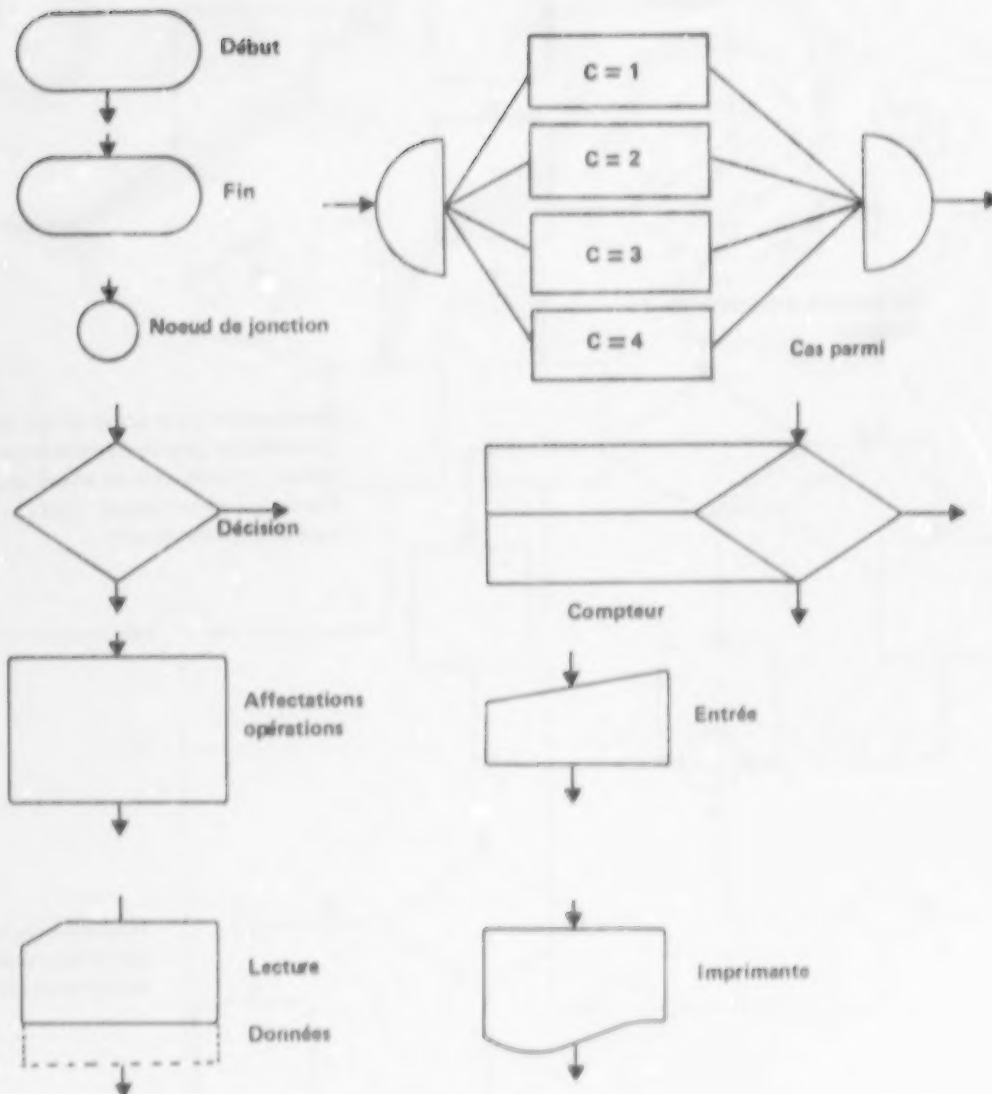
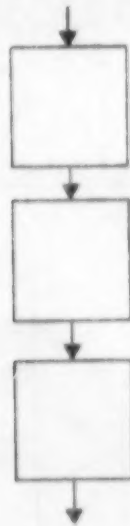


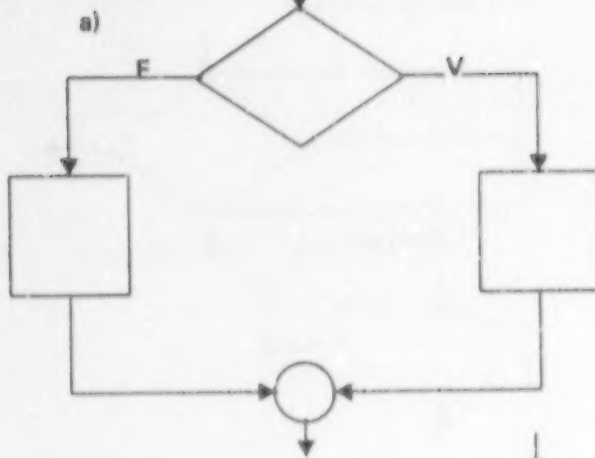
FIG. 1

Structure
Séquentielle



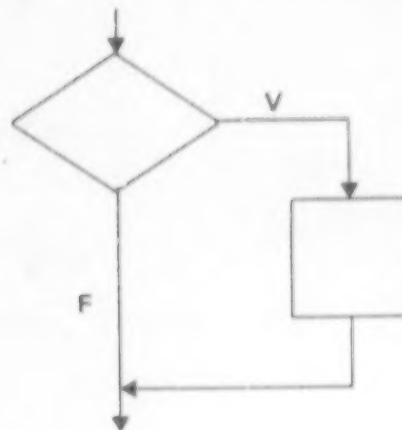
Séquence d'actions que l'ordinateur exécute les uns après les autres sans la possibilité d'en sauter une ou de retourner en arrière.

Structures alternatives
Simples



Branchement à une action plutôt qu'à une autre. La condition peut être exprimée par une expression logique. L'utilisation du noeud de jonction permet d'avoir une sortie unique. Cette forme est aussi appelée sélection binaire.

b)

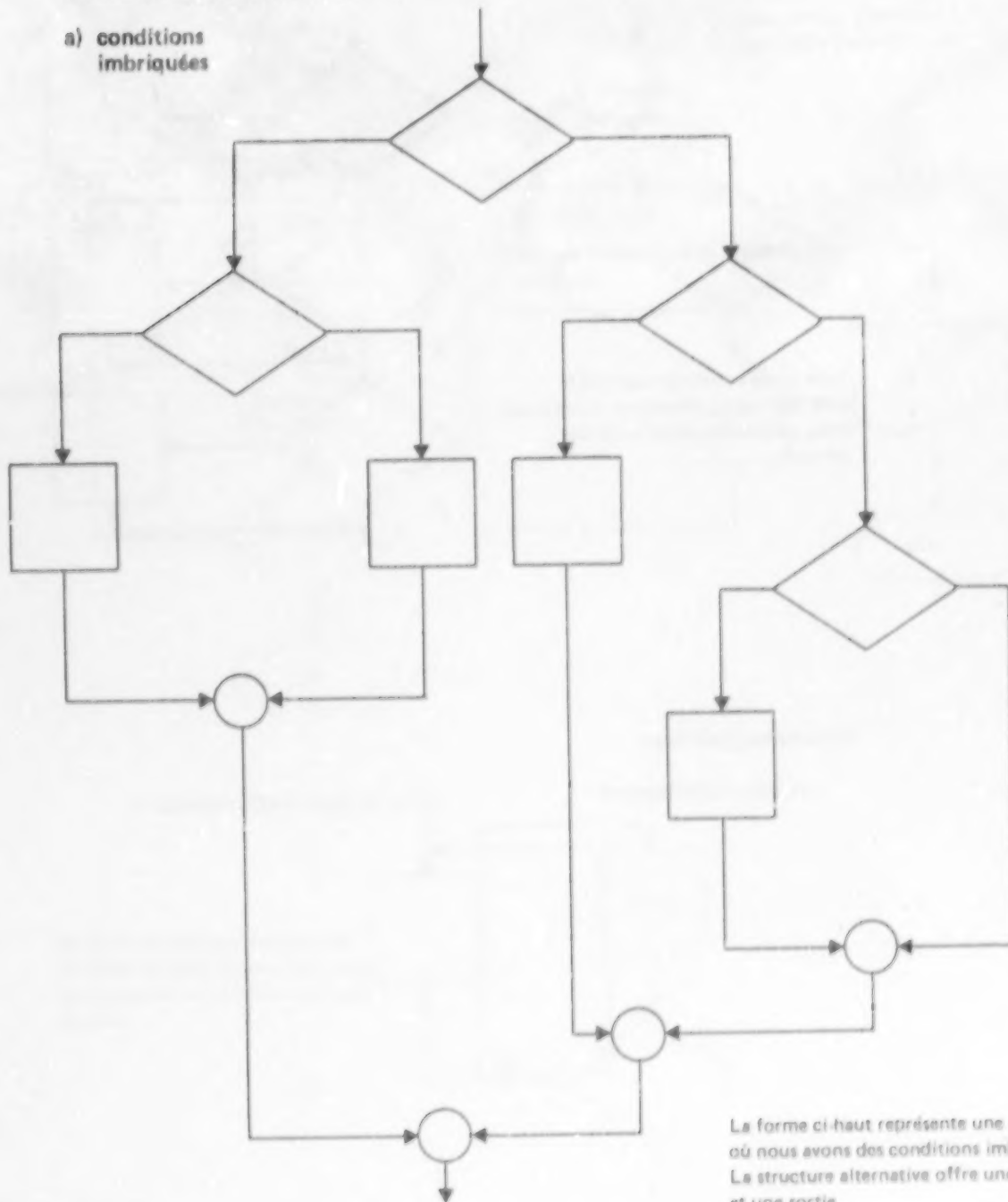


Il peut arriver que l'une des deux parties soit vide. Nous avons la situation suivante :

FIG. 2

Structures alternatives complexes

a) conditions
imbriquées



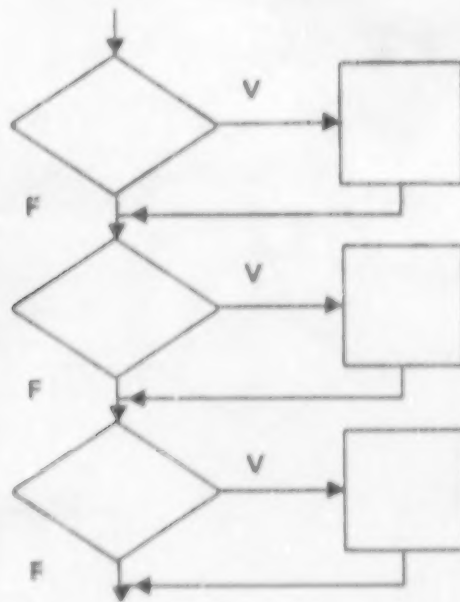
La forme ci-haut représente une situation où nous avons des conditions imbriquées. La structure alternative offre une entrée et une sortie.

FIG. 3

b) enchaînement d'alternative simples

Enchaînement de séquences d'actions

Dans ce cas-ci il est à noter que le SINON n'est pas nécessaire. L'ordinateur passe automatiquement à l'action suivante.



Structures répétitives

— répéter indéfiniment

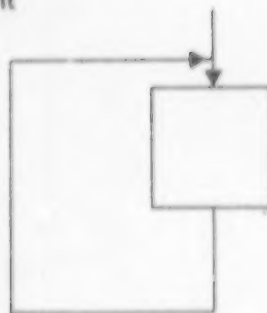
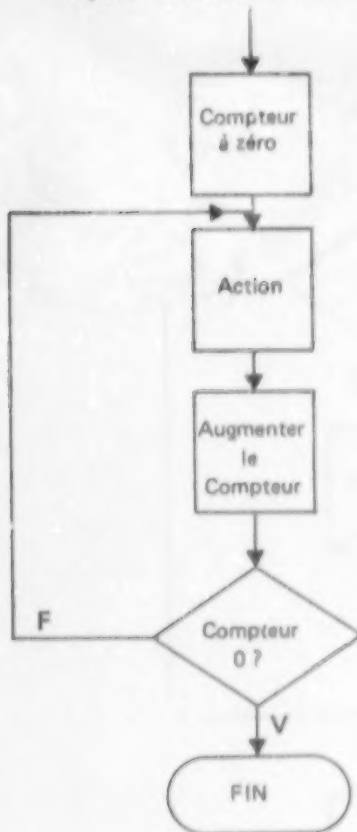
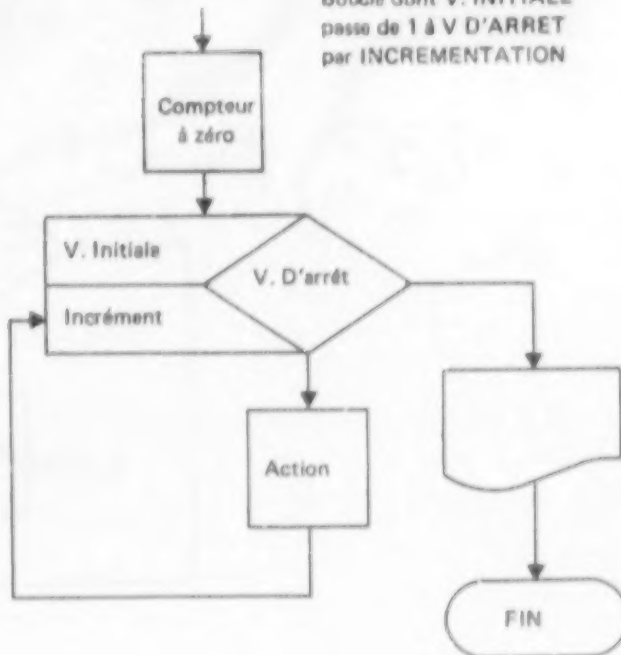


FIG. 4

— répéter un nombre X de fois



Structure itérative
Boucle dont V. INITIALE
passe de 1 à V D'ARRET
par INCREMENTATION



— répéter action jusqu'à ce que

Structure répétitive conditionnelle.
La condition étant placée après l'action
cette dernière est exécutée au moins
une fois.

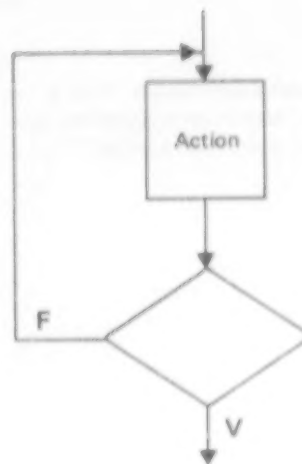
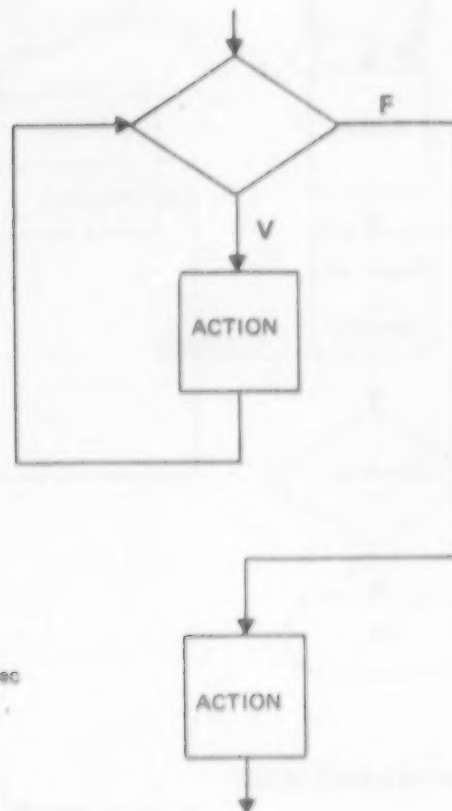


FIG. 5

-- répéter action TANT que



Branchement conditionnel avec utilisation du TANT QUE ...

La condition est placée avant l'action. Il est à remarquer que c'est l'endroit où la condition est placée qui détermine la structure à utiliser.

FIG. 6

ANNEXE 2

PSEUDO-CODE

Concept fondamental en informatique, l'algorithme est décrit comme étant l'ensemble des règles opératoires à suivre pour arriver à la solution d'un problème.

Schématiquement, la méthode algorithmique peut être définie comme suit:

PROBLÈME ANALYSE ALGORITHME CONVERSION PROGRAMME

L'ensemble des expressions utilisées en langage algorithmique, consistant ce que l'on appelle le pseudo-code.

Liste des expressions utilisées

1. DÉCLARATION DES VARIABLES

OBJECTIF: Définir le type et la dimension des variables et tableaux.
Les variables sont de deux types: numérique et chaîne.
On ne précise la dimension que dans le cas des tableaux.

EXEMPLE : Variables numériques X, Y, ALPHA, TABLEAU(20)
Variable chaîne NOM\$
Variable chaîne T\$(20)

2. ÉNONCÉ GÉNÉRIQUE D'ENTRÉE

OBJECTIF: Extraire la valeur des variables au moment de l'exécution.
On ne précise pas le type du périphérique d'entrée.

EXEMPLE : Lire (X, Y, ALPHA)
Lire (NOM\$)

3. ÉNONCÉ GÉNÉRIQUE DE SORTIE

OBJECTIF: Afficher la valeur des variables et des constantes sur un périphérique de sortie

EXEMPLE : Écrire (X, Y, ALPHA)
Écrire ("LA RÉPONSE EST", R)

4. INSTRUCTION D'AFFECTATION

OBJECTIF: Affecter une valeur (constante ou expression) à une variable située à gauche de l'opérateur ().

EXEMPLE : AIRE PI * 2
 X Y
 PRIX 25

5. INITIALISATION DES COMPTEURS ET DES TOTALISEURS

OBJECTIF: Initialiser les valeurs des totalisateurs et des compteurs.

EXEMPLE : Initialiser SOMME à 0
 Initialiser NOMBRE à 0
 Initialiser DERNIER à 100

6. INCRÉMENTATION DES TOTALISATEURS ET DES COMPTEURS

OBJECTIF: Incrémenter ou décrémenter la valeur des accumulateurs, des compteurs et des constantes.

EXEMPLE : Augmenter SOMME de X
 Augmenter COMPTE de 1
 Diminuer SOLDE de X

7. MARQUEUR DE FIN DE PROGRAMME

OBJECTIF: Indiquer la fin d'un programme

EXEMPLE : FIN fin du programme principal
 FIN (RETOUR) fin d'un sous-programme ou procédure

8. MARQUEUR DE DÉBUT DE PROGRAMME

OBJECTIF: Indiquer le début d'un segment de programme (programme principal ou sous-programme) et nommer ce segment.

EXEMPLE : Début

9. INTERRUPTION DE L'EXÉCUTION DU PROGRAMME

OBJECTIF: Interrompre l'exécution d'un programme

EXEMPLE : ARRÊTER

10. ITÉRATION de type "répéter ... jusqu'à ce que"

OBJECTIF: Reprendre l'exécution d'un groupe d'énoncés à plusieurs reprises, jusqu'à la sortie.

EXEMPLE : Répéter
...énoncés
jusqu'à ce que

11. ITÉRATION de type "tant que ... fintant"

OBJECTIF: Reprendre l'exécution d'un groupe d'énoncés à plusieurs reprises, jusqu'à la sortie.

EXEMPLE : Tant que...
...énoncés
Fintant

12. BOUCLE DE PROGRAMME

OBJECTIF: Reprendre l'exécution d'un groupe d'énoncés un nombre donné de fois. Le nombre d'itérations est contrôlé par une variable.

EXEMPLE : Boucle dont INDICE passe de INITIALE à FINALE par PAS
...énoncés...
Fin de boucle

13. ITÉRATION COMPLEXE (boucles imbriquées)

OBJECTIF: Empêcher un certain groupe d'énoncés d'être exécuté lorsqu'une certaine condition est vérifiée--on passe alors à l'itération suivante. Si la condition n'est pas vérifiée, l'itération porte sur tous les énoncés.

EXEMPLE : Boucle dont INDICE passe de 1 à 10
..énoncés..
Boucle dont INDICE passe de 1 à 5
..énoncés..
Fin de boucle
Fin de boucle

14. INSTRUCTION CONDITIONNELLE SIMPLE

OBJECTIF: Permettre l'exécution d'un groupe d'énoncés lorsqu'une condition est vérifiée.

EXEMPLE : SI (condition)
alors
(..groupe d'énoncés..)
FINSI

15. INSTRUCTION CONDITIONNELLE COMPLEXE

OBJECTIF: Permettre l'exécution d'un groupe d'énoncés lorsqu'une expression conditionnelle est vérifiée, et l'exécution d'un autre groupe d'énoncés lorsqu'elle ne l'est pas.

EXEMPLE : SI (condition)
alors
(..énoncés..
sinon
(..énoncés..
FINSI

16. INSTRUCTION À AIGUILLAGES MULTIPLES

OBJECTIF: Permettre la sélection d'un seul traitement parmi plusieurs (au moins trois).

EXEMPLE : Choisir (AIGUILLAGE)
cas 1: (...énoncés...)
cas 2: (...énoncés...)
cas n: (...énoncés...)
autrement: (...énoncés...)

AIGUILLAGE est une variable. Le contrôle est donné à l'instruction correspondant à la valeur de l'expression. Si celle-ci ne correspond pas à une valeur associée à une étiquette, le contrôle est donné à l'énoncé introduit par "autrement". Si la valeur de AIGUILLAGE est fractionnaire, elle sera tronquée. Toutefois, la valeur de AIGUILLAGE n'est pas modifiée. On peut avoir autant d'aiguillages que l'on veut à condition que chacun ait une valeur distincte.

17. DÉFINITION DES FONCTIONS

OBJECTIF: Définir une fonction mathématique à l'aide d'un énoncé algorithmique.

EXEMPLE : Définir la fonction $H(X) = \dots$ expression mathématique..

H représente le nom de la fonction, X est un argument fictif. Sa valeur est locale, par rapport à la fonction.

18. SOUS-PROGRAMMES ou procédures

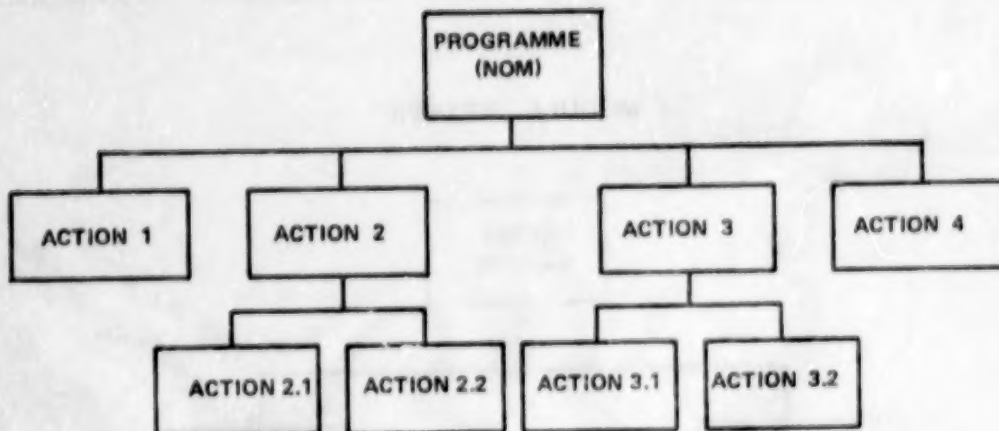
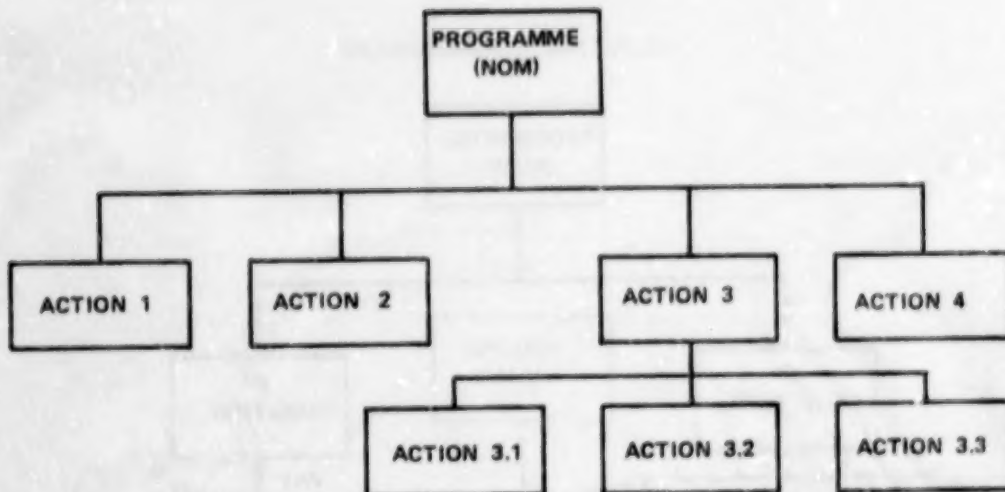
OBJECTIF: Modulariser les programmes au moyen de sous-programmes.

EXEMPLE : Appeler "nom du sous-programme ou procédure"

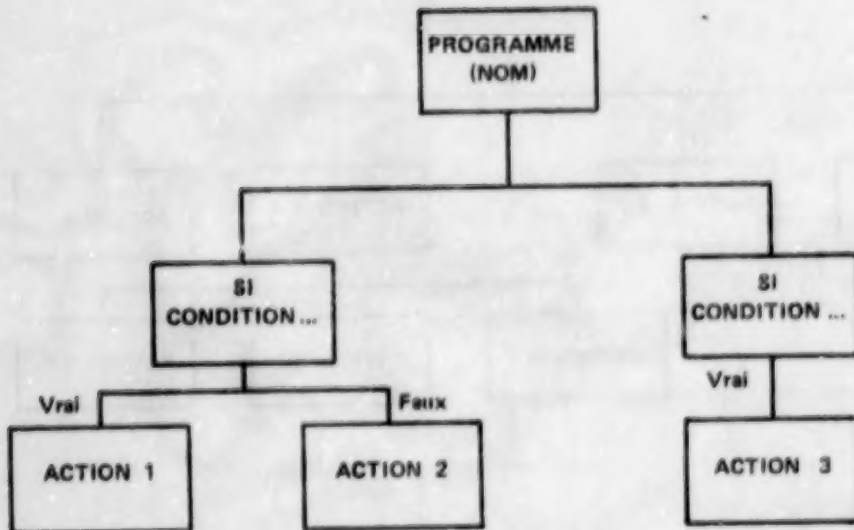
ANNEXE 3

L'ARBRE ALGORITHMIQUE

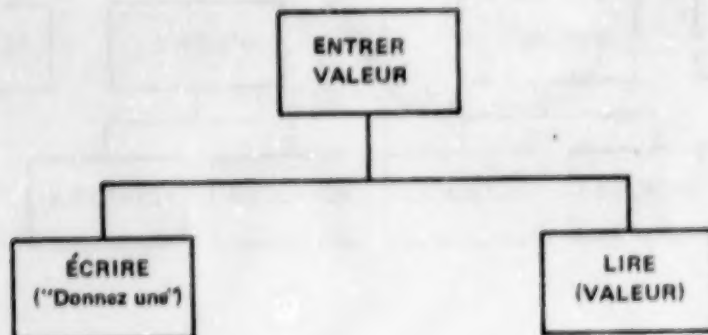
STRUCTURE SÉQUENTIELLE



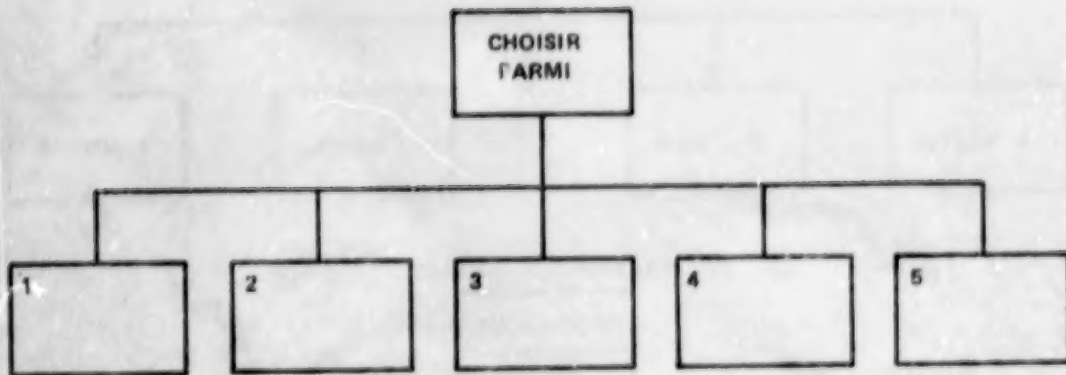
STRUCTURE ALTERNATIVE



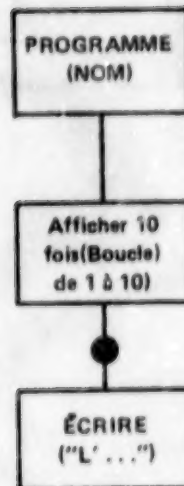
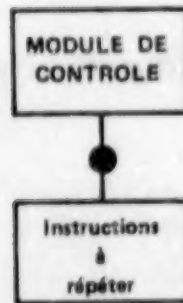
LE MODULE "ENTRER"



BRANCHEMENTS MULTIPLES

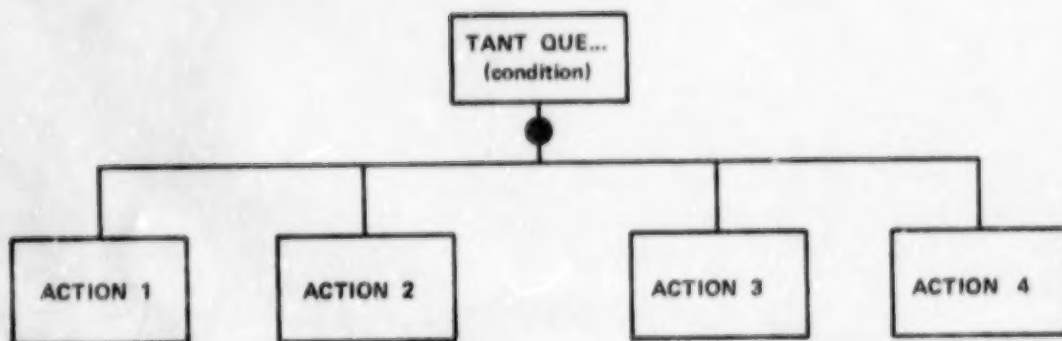


STRUCTURES RÉPÉTITIVES

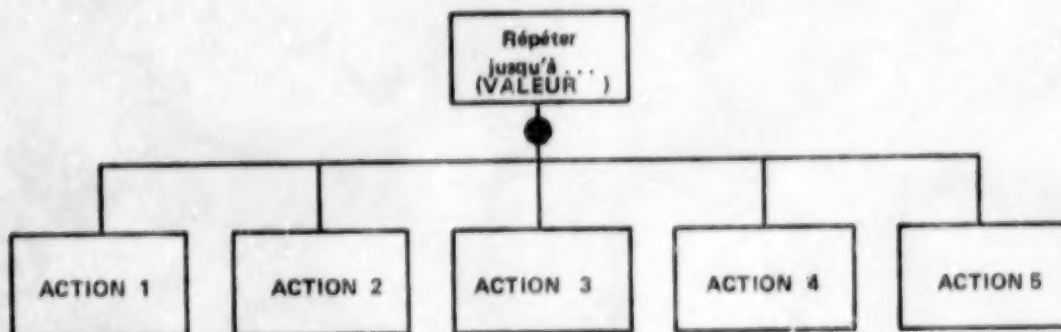


BOUCLE dont VARIABLE passe de _____ à _____ par saut de _____.

STRUCTURES RÉPÉTITIVES



TANT QUE ... FIN TANT



RÉPÉTER ... JUSQU'À ...